НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК_____

«До захисту допущено»
Завідувач кафедри
_____ О.В. Коваль
(підпис) (ініціали, прізвище)
“___”_____2019р.

Магістерська дисертація

зі спеціальності - 122 Комп’ютерні науки
за спеціалізацією - Комп’ютерний моніторинг та геометричне моделювання процесів і систем
на тему
“Інструментальні засоби моделювання сценаріїв аналітики великих даних”

Виконав: студент __6_курсу, групи__ТМ81мп____
Кондрашов Кирило Вадимович_____ _____
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник __доцент к.т.н. Коваль О.В_____ _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент__ доцент к.т.н Сенченко В.Р._____ _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2019

Національний технічний університет України
“ Київський політехнічний інститут ім. Ігоря Сікорського”

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 122 Комп’ютерні науки
за спеціалізацією - Комп’ютерний моніторинг та геометричне моделювання процесів і систем

ЗАТВЕРДЖУЮ
Завідувач кафедри
Коваль О.В. _____
(прізвище, ініціали) (підпис)
«____» _____ 2019р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Кондрашову Кирилу Вадимовичу
(прізвище, ім’я, по батькові)

1. Тема дисертації Інструментальні засоби моделювання сценаріїв аналітики великих даних
Науковий керівник Коваль Олександр Васильович, доцент, к.т.н
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від “_04__”__листопада___2019__року № 3812-с
2. Строк подання студентом дисертації 10 12 2019
3. Об’єкт дослідження Сценарії аналітики великих масивів даних
4. Предмет дослідження Інструментальні засоби побудови сценаріїв аналітики великих даних
5. Перелік питань, які потрібно розробити
   1. Аналіз існуючих систем засобів для побудови сценаріїв аналітики великих даних.
   2. Аналіз технологій для роботи із великими даними.
   3. Розробка інструментальних засобів для побудови сценаріїв аналітики великих даних на базі проведених вище аналізів
   4. Розробка стартап-проекту

6.Орієнтовний перелік графічного (ілюстрованого) матеріалу презентація на тему “Інструментальні засоби моделювання сценаріїв аналітики великих даних”

7. Орієнтований перелік публікацій
   1. Кондрашов К. Інструментальні засоби побудови сценаріїв аналітики великих даних в інформаційно-аналітичних системах // Інтеграція світових наукових

процесів як основа суспільного прогресу : Матеріали III Міжнародної науково-практичної конференції (м. Київ, 22–23 листопада 2019 р.)

8. Дата видачі завдання «10» вересня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	строки виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	10.09	
2	Огляд існуючих рішень	11.09 – 31.03	
3	Огляд існуючих технологій	1.04 – 24.08	
4	Розробка власного програмного забезпечення	25.08 – 13.10	
5	Тестування	14.10 – 25.10	
6	Оформлення дипломної роботи	26.10 – 9.12	
7	Отримання допуску до захисту та подача роботи в ДЕК	10.12	

Студент _____ Кондрашов К.В.
(підпис) (прізвище та ініціали)

Науковий керівник _____ Коваль О.В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ

виконана на тему “Інструментальні засоби моделювання сценаріїв аналітики великих даних”

Структура та обсяг дипломної роботи. Магістерська дисертація складається зі вступу, чотирьох розділів, висновку, переліку посилань з 50 найменувань, 1 додаток, і містить 33 рисунки, 23 таблиць. Повний обсяг магістерської дисертації складає 97 сторінок.

Актуальність теми. Стрімкий розвиток інформаційних технологій та комп’ютеризація усіх аспектів діяльності людини в сучасному суспільстві призводить до надмірного росту генерації нових даних. Зі збільшенням об’єму інформації постає питання про необхідність пошуку нових способів та методів зберігання, репрезентації, систематизації, формалізації тощо.

Тому все більше компаній з усього світу починають цікавитися програмним забезпеченням, яке б надало змогу використовувати переваги нових технологій роботи із великими даними у простій та зрозумілій формі.

Мета дослідження полягає в розробці інструментальних засобів моделювання сценаріїв аналітики великих даних.

Для досягнення поставленої задачі були сформульовані наступні **завдання дослідження**, що визначили логіку дослідження та його структуру:

- проаналізувати існуючі інструментальні засоби моделювання сценаріїв аналітики великих даних;
- проаналізувати існуючі технології для роботи великих даних;
- створити власні інструментальні засоби моделювання сценаріїв аналітики великих даних з урахуванням недоліків існуючих систем.

Об’єктом дослідження є аналіз способів і засобів обробки великих даних у прикладних системах.

Предметом дослідження є інструментальні засоби формування сценаріїв аналітики великих даних.

Наукова новизна одержаних результатів. Найбільш суттєвими науковими результат полягає в тому, що була створена система, яка надає необхідний інструментарій для аналізу великих даних у графічній формі із гнучкими налаштуваннями методів та алгоритмів із повністю відкритим вихідним кодом.

Практичне значення одержаних результатів. Сформульовані основні концепції, на які потрібно звернути увагу при проектуванні інструментальних засобів сценарії аналітики великих даних, описані основні методи та підходи, які були використані.

Публікації. Кондрашов К. Інструментальні засоби побудови сценаріїв аналітики великих даних в інформаційно-аналітичних системах // Інтеграція світових наукових процесів як основа суспільного прогресу : Матеріали III Міжнародної науково-практичної конференції (м. Київ, 22–23 листопада 2019 р.)

Ключові слова. *Інструментальні засоби моделювання сценаріїв, сценарій, великі дані, інтелектуальний аналіз даних.*

ABSTRACT

ON MASTER'S THESIS

on topic "Big Data analytic scenarios modelling tools"

Topicality The master's thesis consists of an introduction, four sections, a conclusion, a list of links of 50 titles, 1 appendix, and contains 33 figures, 23 tables. The full volume of the master's thesis is 97 pages.

Purpose. The rapid development of information technology and the computerization of all aspects of human activity in today's society leads to the excessive growth of data generation. With the increasing volume of information, the question arises of the need to find new ways and methods of storage, representation, systematization, formalization and more.

That's why more and more companies around the world are beginning to become interested in software that can take advantage of new big data technologies in a simple and understandable way.

The purpose of the study is to develop tools for modelling an analytic scenario of Big Data

To achieve this task, the following **research objectives** were formulated:

- analyse existing tools for modelling big data analytics scenarios;

- analyse existing technologies for big data;

- create my own big data analytics scripting tools, taking into account the weaknesses of existing systems.

The object of research is to analyze the ways and means of processing big data.

The subject of research is tools for modelling Big Data analytics scenarios.

Scientific novelty. The most significant scientific result is that a system has been created that provides the necessary tools to analyse big data graphically with flexible, open source methods and algorithms.

The practical value of research. The main concepts that should be considered when designing Big Data Analytics tools are outlined, the main methods and approaches that have been used are described.

Publications. Kondrashov K. Instrumental tools for building big data analytics scenarios in information-analytical systems // Integration of world scientific processes as the basis of social progress: Proceedings of the III International Scientific and Practical Conference (Kyiv, November 22-23, 2019)

Key words. *Big Data analytic scenarios modelling scenario, big data, data mining.*

ЗМІСТ

ВСТУП 10
1. ДОСЛІДЖЕННЯ ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ ПОБУДОВИ СЦЕНАРІЇВ АНАЛІТИКИ ВЕЛИК 13
1.1. Великі дані та їх актуальність у сучасному світі 13
1.2. Основні характеристики та методи, що використовуються при побудові сценарію аналітики 16
1.3. Аналіз існуючих систем для побудови сценаріїв аналітики великих даних 19
1.3.1 Система Rapid Miner 20
1.3.2 Система Orange 21
1.3.3 Система KNIME 23
1.3.4 Система WEKA 24
1.3.5 Система Alteryx 25
1.4. Висновки до розділу 1 26
2. РОЗРОБКА ІНТСРУМЕНТАЛЬНИХ ЗАСОБІВ ПОБУДОВИ СЦЕНАРІЇВ АНАЛІТИКИ ВЕЛИКИХ ДАНИХ 27
2.1. Вибір мови програмування 27
2.1.1 Мова програмування Java 28
2.1.2 Мова програмування Scala 29
2.1.3 Мова програмування R 31
2.1.4 Мова програмування Python 33
2.1.5 Висновки порівняльного аналізу мов програмування 34
2.2 Вибір бібліотек мови програмування Python для побудови сценаріїв аналітики великих даних 35
2.3 Формування функціоналу інструментальних засобів моделювання побудови сценаріїв аналітики великих даних 37
2.3.1 Віджети для зчитування даних 38

2.3.2 Віджети для маніпуляції із даними 42
2.3.3 Віджети для попередньої обробки даних 46
2.3.3 Віджети для візуалізації 47
2.3.3 Віджети для інтелектуального аналізу даних 49
2.4. Висновки до розділу 2 56
3. ПРИКЛАД РОБОТИ ІНТСРУМЕНТАЛЬНИХ ЗАСОБІВ ПОБУДОВИ СЦЕНАРІЇВ АНАЛІТИКИ ДЛЯ ВЕЛИКИХ ДАНИХ 57
3.1. Задача на регресію 57
3.2. Задача на класифікацію 61
3.3 Висновки до розділу 3 65
4. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ “ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ПОБУДОВИ СЦЕНАРІЇВ АНАЛІТИКИ ВЕЛИКИХ ДАНИХ” 66
4.1. Опис ідеї проекту 66
4.2. Технологічний аудит проекту 69
4.3. Аналіз ринкових можливостей 70
4.4. Розробка ринкової стратегії проекту 77
4.5. Розробка маркетингової програми 80
4.6. Висновки до розділу 4 84
ВИСНОВКИ 85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 87
Додаток 1. Апробація 92

ВСТУП

Стрімкий розвиток інформаційних технологій та комп'ютеризація усіх аспектів діяльності людини в сучасному суспільстві призводить до надмірного росту генерації нових даних. Зі збільшенням об'єму інформації постає питання про необхідність пошуку нових способів та методів зберігання, репрезентації, систематизації, формалізації тощо.

І з кожним роком актуальність зазначеного питання лише збільшується, оскільки згідно с дослідженнями проведеними у 2017 році кожен день людство генерує два с половиною квінтильйона байт у різних формах — від купівлі нового телефона у інтернет магазині до банального відправлення електронних листів через сервіси Google. У зв'язку з надмірним ростом інформації та розумінням, що класичні методи та системи управління базами даних не здатні ефективно обробляти її все активніше набирає популярність термін "великих даних" переклад з англійського терміну "big data".

У широкому сенсі великі дані можна розглядати, як соціально-економічний феномен оскільки нові технологічні можливості надали змогу кардинально змінити підходи для ведення аналітичної діяльності. Під аналітичною діяльністю можна вважати сукупність певного переліку дій на основі методів та засобів для пошуку, зберігання, обробки, аналізу та представлення даних з метою прийняття рішень на управлінському рівні. Важливою складовою такої діяльності є сценарний аналіз, який досліджує наскільки істотним є вплив конкретного переліку факторів. Наприклад, застосовуючи такий підхід компанія може набагато легше обробляти інформацію із різноформатних систем приводячи до спільного знаменника, створювати більш якісні передбачення, що призведе до все стороннього розуміння потреб користувачів. Так провайдери послуг телекомунікації зможуть більш ефективно приваблювати нових клієнтів та підтримувати інтерес вже існуючої бази. Продуктові компанії зможуть

більш якісно розуміти не тільки які товари продаються краще або гірше, а чому так відбувається і які зміни необхідно вносити до існуючої стратегії.

Тому все більше компаній з усього світу починають цікавитися програмним забезпеченням, яке б надало змогу використовувати переваги нових технологій роботи із великими даними у простій та зрозумілій формі.

У дипломній роботі буде вирішуватися задача створення інструментальних засобів моделювання сценарії великих даних. Так як для роботи із інформацією потрібно проводити багато маніпуляцій даними на різних етапах формування сценарію, то багатофункціональний візуальний редактор значно спростить роботу аналітиків, яким не потрібно буде володіти жодною мовою програмування та витрачати час на написання коду.

Поставлена мета вимагає вирішення наступних наукових задач:

- проведення аналізу порівняння існуючих програмних рішень та виділення основних переваг та недоліків;
- проведення аналізу порівняння існуючих технологій для роботи із великими даними та вибір кращого із представлених;
- дослідження методів очистки та обробки даних;

Метою дипломної роботи є розробка такого програмного забезпечення, яке буде надавати можливість побудови сценаріїв аналітики великих даних. Важливо проаналізувати підходи до вирішення таких завдань, щоб обрати найефективніший підхід, який би підійшов для подібної задачі.

Об’єктом дослідження є аналіз способів і засобів обробки великих даних у прикладних системах, які доступні у відкритому доступі. Предметом дослідження є інструментальні засоби формування сценаріїв аналітики.

Досягнення поставленої мети реалізовано з використанням мови програмування Python, що є однією із передових мов у сфері big data. Для реалізації графічного інтерфейсу було використано фреймворк PyQt п’ятої версії. Для безпосередньої роботи із даними були обрані бібліотеки scikit-learn, pandas та numPy.

Наукова новизна дипломної роботи полягає в тому, що була створена система, яка надає необхідний інструментарій для аналізу великих даних у графічній формі із

гнучкими налаштуваннями методів та алгоритмів із повністю відкритим вихідним кодом.

Потенційні застосування та практична цінність результатів дипломної робити:

1. Сформульовані основні концепції, на які потрібно звернути увагу при проектуванні інструментальних засобів сценарії аналітики великих даних, описані основні методи та підходи, які були використані;
2. Оскільки програмне забезпечення реалізоване на мові програмування Python то подальша підтримка та розширення системи не буде дуже складною задачею. Реалізація нового та корегування вже існуючого функціоналу відбувається доволі швидко, що дозволить підлаштовувати систему під різні проблемні області;
3. Візуалізація дозволяє більш наглядно продемонструвати результати і наслідки, та корегувати стратегії розвитку підприємства.

1. ДОСЛІДЖЕННЯ ІСНУЮЧИХ МЕТОДІВ ТА ЗАСОБІВ ПОБУДОВИ СЦЕНАРІЇВ АНАЛІТИКИ ВЕЛИКИХ ДАНИХ

1.1 Великі дані та їх актуальність у сучасному світі

Якщо розглядати термін “великі дані” з технічної точки зору, то він має декілька інтерпретацій, і одна з них означає позначення великих груп як структурованих так і неструктурованих даних величезних обсягів різної форми, які неможливо обробляти традиційними способами та підходами[1]. Великі дані повинні бути оброблені за допомогою сучасних методів аналітики та алгоритмів для виявлення значущої інформації. В такому розумінні термін великі дані з’явився у вересні 2008, коли редактор великобританського наукового журналу Nature Кліффорд Лінч підготував статтю “Як можуть вплинути на майбутнє науки технології, що відкривають можливість роботи із великими об’ємів даних?”. В матеріалі йшла мова про неочікуване явище вибухового зростання обсягу та різноманітності інформації, яке людство почало генерувати за допомогою діяльності комп’ютерів.

Основні характеристики великих даних було записано в класичному вигляді за допомогою трьох V:

- volume (обсяг) — згенерованої та збереженої інформації, що дає представлення рахувати якийсь набір даних великими чи ні;
- variety (різноманіття) — типів даних, яких зберігають;
- velocity (швидкість) — під цим терміном розуміється одразу дві значення. Перша швидкість — це скільки генерується нової інформації за проміжок часу. Друга — скільки витрачається часу на обробку певного кількості байт.

Із часом та розвитком напрямку big data додавались додаткові характеристики, такі як veracity (достовірність)[2], viability (життєздатність), value (цінність) та visualization (візуалізація), проте саме ці три V можна вважати основними. Більш

детальне представлення як кожна V-характеристика залежить один від одної можна побачити на рисунку 1.1.

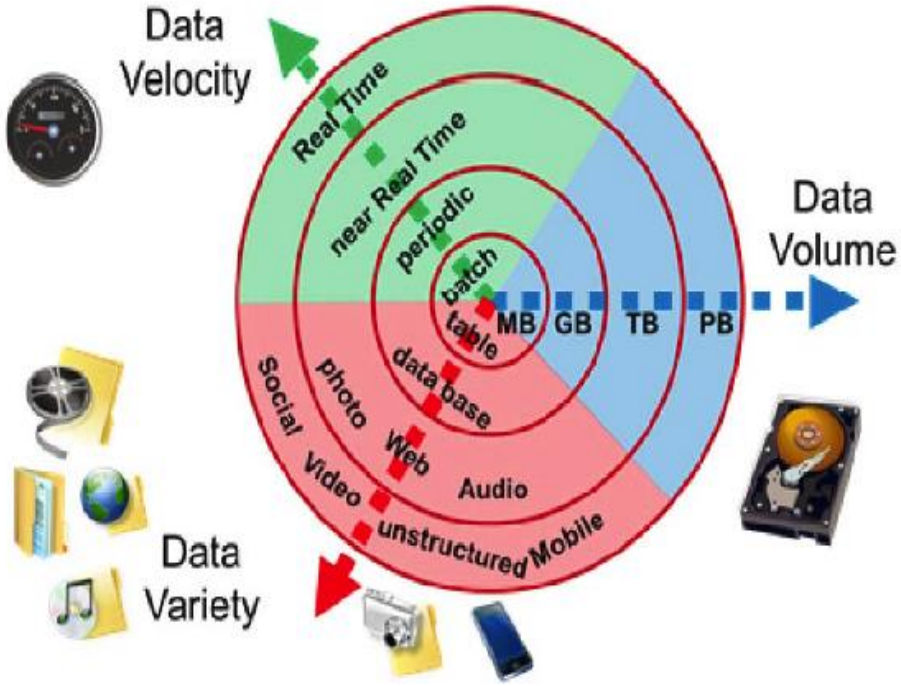


Рисунок 1.1 — Ріст основних характеристик великих даних

Альтернативне тлумачення терміну полягає в тому, що big data — це набір інструментів та методів прогностичної аналітики, аналітики поведінки користувачів або деяких інших сучасних методів аналізу даних, які витягують значення з інформації, яка міститься у дата сетах різних за розмірами. Такі технології кардинально змінюють підходи аналізу великих об’ємів даних[3], оскільки дозволяють знайти та розпінити кореляції, яких раніше не було помітно.

Набор інформації, який може зберігатися в дата сетах ділиться на три основні категорії:

- структурована (таблиці SQL, Excel файли);
- неструктурована (текс, аудіо данні, відео данні, зображення);
- слабоструктурована (XML, SJON).

Великі дані проникають у кожну сферу нашого життя і надають змогу робити революційні речі. Так наприклад, обробка більше ніж п’яти петабайт інформації зі супутників надали змогу отримати приблизне зображення чорної діри в галактиці під кодовою назвою Messier 87, що знаходиться на відстані в п’ятдесяти п’яти мільйонів

світлових років від нашої галактики. На збір необхідної інформації знадобилося більше двох років[4].

Іншим визначним прикладом може слугувати американський стрімінговий сервіс Netflix. Аналізуючи здавалось би не пов’язані між собою інформацію о користувачах таку як:

- локацію користувача;
- взаємодію між користувачем та відео (паузи, повторний перегляд тощо);
- девайс на якому користувач дивився контент;
- пошукові запити на платформі;
- інформацію з Twitter;
- інформацію з Facebook;
- оцінка на платформі.

Завдяки цьому Netflix персоналізує контент та розуміє, який фільм/серіал буде цікавий користувачам та у які роботи потрібно вкладати гроші, а які можна ігнорувати. Усе це призводить до кращого розуміння запитів користувача і підвищує шанси, що той залишиться і продовжить підписку. Завдяки такому підходу дохід американської кампанії зріс з двох мільярдів до ста семи десяти мільярдів доларів США за менше ніж чим десять років[5].

Тому зацікавленість сучасних кампаній та конгломератів у програмному забезпечені, яке б надавало можливість створювати сценарії, дивитися на можливі прогнози, виявляти скриті закономірності, візуалізувати їх та корегувати маркетингові стратегії в залежності від отриманих результатів. Тепер людство може розуміти не тільки “що” відбулось, а “чому” відбулось використовуючи великі дані.

Більш того, такі системи знизили б необхідність бізнес аналітикам освоювати такі мови програмування, як Python, Scala або Java, що широко використовуються у сфері великих даних. Тобто не потрібно буде витрачати час на написання коду, подальше його віддладку і тестування отриманих результатів, що може зайняти багато часу. Розглядаючи системи побудови сценарії аналітики великих даних треба конкретизувати, що включає в собі термін “побудова сценарію” і який функціонал необхідно реалізувати в кінцевому програмному продукті, для того що б створити

конкуренто спроможній стартап-проект на ринку інструментальних засобів побудови сценаріїв аналітики.

1.2 Основні характеристики та методи, що використовуються при побудові сценарію аналітики

Основна мета аналітичної діяльності полягає у забезпеченні інформаційних потреб підприємства та підтримка при прийнятті рішень. Аналітична діяльність під собою розуміє роботу із накопиченою інформацією: обробку, аналіз, візуалізацію у формі різних звітів, підтвердження результатів з боку експертної групи та отримання обґрунтованих та якісних рішень на базі згенерованих знань. Під сценарієм аналітики можна вважати певну послідовність кроків, шо виконує аналітик або група аналітиків при вирішенні задач[6].

Будь-який сценарій можна представити як ітераційну послідовність ряду певних етапів роботи із інформацією. Для того щоб виділити основні кроки, характерні для побудови будь-якого аналітичного сценарію для роботи із великими даними можна розглянути наступний рисунок 1.2.

Рисунок 1.2 — Основні етапи побудови сценаріїв аналітики великих даних

Розглядаючи з точки боку аналітика сценарій складається з п’яти етапів, які необхідно розглянути більш детально для розуміння функціоналу, який потрібно реалізувати:

- Identifying problem, в перекладі ідентифікація проблеми. Це найперший етап оскільки дає зрозуміти бізнес аналітику, які саме процеси на підприємстві функціонують не так, або не достатньо ефективно, є потенційні втрати доходу. Також необхідно виділити зміст та необхідні ресурси для побудови сценарію. Цей етап має мінімальне відношення до функціоналу напряму, а більше до здібностей самого аналітика;
- Designing Data Requirement, тобто етап на якому бізнес аналітик готує інформацію із різних систем, які використовує підприємство і яку може бути релевантна при побудові сценарію. Чим більше засобів для завантаження даних у різних розширеннях, від Excel файлів до інформації із серверу у форматі JSON, може запровадити система для аналітики, тим менше підготовчої роботи необхідно буде виконати людині;
- Pre-processing Data, це етап на якому відбувається попередня обробка завантажених даних. Це один із найважливіших моментів при роботі із великими даними, оскільки забезпечення широкої репрезентації та якості інформації це першочергове завдання перед будь-яким аналізом.[7] Під широким терміном попередньої обробки даних розуміють очистку даних, їх трансформацію, редукцію та нормалізацію. Окремо про це буде розглянуто далі в роботі, оскілки ці методи необхідно розписувати більш детально;
- Performing Analytics Over Data в перекладі виконання аналітики над отриманими даними. Це основний момент при побудові сценаріїв, який необхідно реалізовувати в системі. Оскільки саме від представлення кількості алгоритмів та методів, а також можливості налаштовувати їх параметрів під різні непередбачувані ситуації дозволяє будувати сценарії різної степені складності та ефективності;
- Visualizing Data, тобто візуалізація отриманих результатів. При роботі бізнес аналітику необхідно проаналізувати та правильно представити отриману інформацію перед керівництвом або іншим відділам компанії. Проте вони можуть не

мати достатнього досвіду для розуміння великого набору даних у формі таблиць, тощо. Просто набір цифр і пояснювальний текст буде не настільки ефективним, як простий графік, що вказує на потенційні проблеми більш наглядно[8].

З набуттям популярності напрямку роботи із великими даними, нові алгоритми та методи дозволили змінити класичні підходи до створення сценаріїв та розробити нові напрямки надаючи на виході інформацію, яка раніше неможливо було отримати. Серед них можна виділити наступні:

- Descriptive Analytics Scenarios (сценарії описової аналітики). Такий підхід використовує агрегацію вхідних даних та використовує технології інтелектуального аналізу (data mining), щоб забезпечити розуміння минулого та надати відповідь на питання: «Що трапилось?» Описова аналітика при побудові сценарію робить саме те, що випливає з назви, вона «описує», або узагальнює вихідні дані та робить їх легче для інтерпретації людьми[9];
- Predictive Analytics Scenarios (сценарії аналітики передбачення). Цей підхід використовує статистичні моделі та методи прогнозування, щоб зрозуміти майбутнє та відповісти на питання: «Що може статися?» Прогнозна аналітика, яка використовується в рамках побудови сценарію, надає компаніям ділову інформацію на основі накопичених даних та дає ймовірнісні оцінки майбутнього результату[10];
- Prescriptive Analytics Scenarios (сценарії прописної аналітики). Цей підхід використовує алгоритми оптимізації та моделювання для отримання порад щодо можливих результатів та відповідей на питання: «Що робити?» Це дозволяє користувачам «прописати» ряд різних можливих дій, оцінити ефективність та втілити їх у житті для вирішення існуючих проблем. Ця аналітика полягає в наданні порад і є однією з найскладніших[11];
- Diagnostic Analytics Scenarios (сценарії аналітики діагностики). Цей підхід використовується для визначення того, чому щось сталося в минулому. Він характеризується такими методами, як деталізація, інтелектуальний аналіз даних та кореляція. Діагностична аналітика більш глибоко розглядає дані, щоб зрозуміти першопричини подій[12].

Мета будь-якого із вище зазначених типів сценаріїв на основі математичних та статистичних принципів забезпечити прогнози, проаналізувати ризики або вказати на слабкі місця у структурі підприємства або ведення справ починаючи від результатів великої ймовірності до дуже малоймовірних. Отже, інструментальні засоби для побудови сценарії повинні забезпечувати наступні можливості:

- гарантувати можливість будувати різні типи сценаріїв;
- надавати можливість змінювати параметри для отримання різних результатів в межах одного сценарію;
- надавати оцінку точності отриманих результатів.

Отже розглянувши основні етапи побудова сценаріїв аналітики великих даних можна зрозуміти, який функціонал потрібно реалізувати. Наступним кроком при написані магістерської дисертації буде розгляд існуючих рішень.

1.3 Аналіз існуючих системи для побудови сценаріїв аналітики великих даних

З ростом популярності тренду великих даних та необхідності проводити аналіз із застосуванням програмного забезпечення, доступних засобів для цього на ринку стає все більше. Саме для цього необхідно провести детальний аналіз існуючих рішень та виділити основні переваги та недостати. Основними критеріями, які необхідно розглянути при виборі кандидатів наступні:

- графічний інтерфейс повинен реалізовувати функціонал drag and drop (перетягування). Це забезпечує простоту в освоєнні інструменту для бізнес аналітика та можливість наглядно переставляти сценарії різної степені складності у простій візуальній формі;
- функціонал програмного забезпечення повинен повністю забезпечувати усі етапи побудови сценаріїв аналітики розглянуті в минулому розділі;
- методи та алгоритми обробки великих даних, застосовані в програмному забезпечені повинні мати гнучкі параметри налаштування.

Для пошуку були використано сайт G2 Crowd (https://www.g2.com) який є одним із найбільших сайтів для оцінювання програмного забезпечення. Серед усіх інструментів для аналітики великих даних були обрані наступні засоби:

- система Rapid Miner;
- система Orange;
- система Weka;
- система Knime;
- система Alteryx.

1.3.1 Система Rapid Miner

Система Rapid Miner — це кросплатформене програмне забезпечення, що розроблене німецькою компанією з однойменною назвою у 2006 році. Rapid Miner надає гнучке середовище для машинного навчання, дата майнінгу та інтелектуального аналізу тексту. Програмне забезпечення використовується, як у інтернаціональних компаніях для аналітики великих бізнес процесів, так і для наукових дослідженнях в університетах та інститутах, навчання студентів, швидкого створення прототипів і розробки додатків. Rapid Miner розроблений на основі клієнт-серверної модель і має, як платну так і безкоштовну версії. Написаний Rapid Miner на мові програмування Java та має відкритий код на сервісі GitHub. Серед обраного програмного забезпечення функціонал, що надає Rapid Miner можна назвати одним із найкращих, оскільки надає найбільш гнучкий інструментарій.

Для роботами з різними джерелами даних у Rapid Miner існує близько 53 операторів представлених на рисунку 1.3.

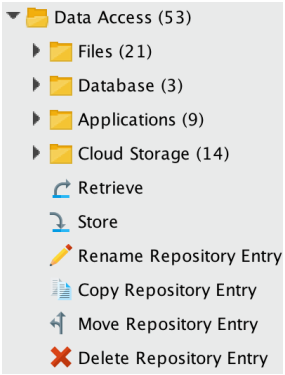


Рисунок 1.3 - Оператори “Data Access” у Rapid Miner

Серед них є оператори для роботи із файлами різного формату (csv, excel), програмним забезпеченням (Salesforce, Twitter), реляційними базами даних, хмарними сховищами такі як Amazon, Google і Azure, а також локальними репозиторіями. Проте робота із веб сервісами не доступні у безкоштовній версії даного ПЗ.

Більш того в безкоштовній версії Rapid Miner можливо працювати лише із набором даних, кількість рядків у яких не перевищує 10 000. Не можна також використовувати технологію Radoop, що являє собою інтеграцію файлової системи Hadoop у функціонал Rapid Miner.

Для процесу агрегації даних представлено більше ніж 76 операторів серед яких є фільтрування, оператори join, merge тощо. Для очистки даних надається більше ніж 25 операторів. Для побудови сценаріїв аналітики великих даних надається більше 153 оператори, серед яких різні типи регресії (логістичні, лінійні) та класифікацій.

Більш того, у Rapid Miner вбудовані оператори, що забезпечують процес валідації та оцінюють якість отриманого результату. Окрім цього візуалізувати результат можна у різних графічних форматах із різними налаштуванням, які можна зберігати у форматі jpg, png та pdf.

1.3.2 Система Orange

Система Orange — це компонентний набір кросплатформеного програмного забезпечення для побудови сценаріїв аналітики великих даних різної складності, що містить інтерфейс, що підтримує функціонал drag and drop для дослідження та візуалізації даних. Розвиток програмного продукту розпочався в 1997 році в лабораторії біоінформатики в Люблінському університеті. Для роботи із даними в Orange за замовчуванням є три оператори, що представлені на рисунку 1.4:

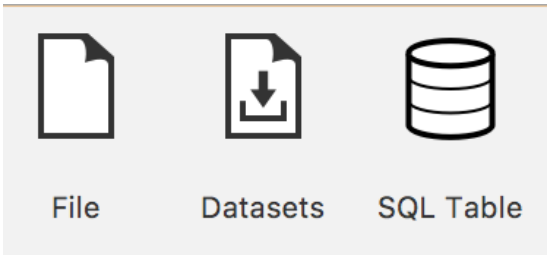


Рисунок 1.4 - Оператори для роботи із даними в Orange

Оператор “File” працює із файлами, що підтримують формат csv. Оператор “Datasets” підтримує дані, що були завантажені із онлайн репозиторія Orange. Оператор SQL Table дозволяє завантажити дані із серверу PostgreSQL та SQL Server. Особливість однак полягає в тому, що для цього необхідно вручну прописати завантаження необхідних бібліотек мови Python, що робить процес налаштування Orange значно складніше.

Для того, щоб мати змогу працювати із текстовими документами, зображеннями, тощо необхідно завантажувати окремі пакети, що знов ускладнює процес роботи із Orange. Більш того налаштування та поведінка деяких операторів Orange не дуже гнучке, що може значно зменшити цінність отриманого в ході аналізу результатів.

Функціонал Orange за замовчуванням помітно менший ніж у Rapid Miner. Для очистки даних використовується лише один оператор Preprocess, що дозволяє нормалізувати данні, опрацьовувати рядки із відсутніми значеннями, тощо. Для побудови прогнозу є всього шістнадцять операторів, які тим не менш покривають основний мінімум — є лінійна та логістична регресії, різні типи класифікації.

1.3.3 Система KNIME

Система KNIME — це кросплатформене програмне забезпечення для побудови сценаріїв аналітики великих даних з частково відкритим вихідним кодом, розроблена та підтримувана однойменною компанією. Розвиток KHIME розпочався у січні 2004 року командою програмних інженерів Університету м. Констанц. На чолі команди на той час був Майклом Бертолдом, а самі програмісти прийшли з компанії в Силіконовій долині, яка робила додатки для фармацевтичної промисловості. Початковою метою було створення модульної, високо масштабованої та відкритої платформи інтелектуальної обробки даних не орієнтуючись на якусь конкретну область застосування. Програмне забезпечення має як безкоштовну версію, так і платну. Мова програмування на якій реалізована KNIME — Java.

KNIME може працювати із наступними джерелами даних:

- формат Xlsx;

- формат ARFF;
- формат XML;
- формат JSON;
- формат SQL Server;
- формат MongoDB;
- формат REST сервери (лише у платній версії).

Надаваємий KNIME функціонал покриває задачу попередньої обробки даних, хоча кількість операторів менша ніж у Rapid Miner. Проте операторів для прогнозу, який нараховується сто одиниць, цілком достатньо щоб побудувати сценарій високої складності. Більш детально категорії представлено на рисунку 1.5

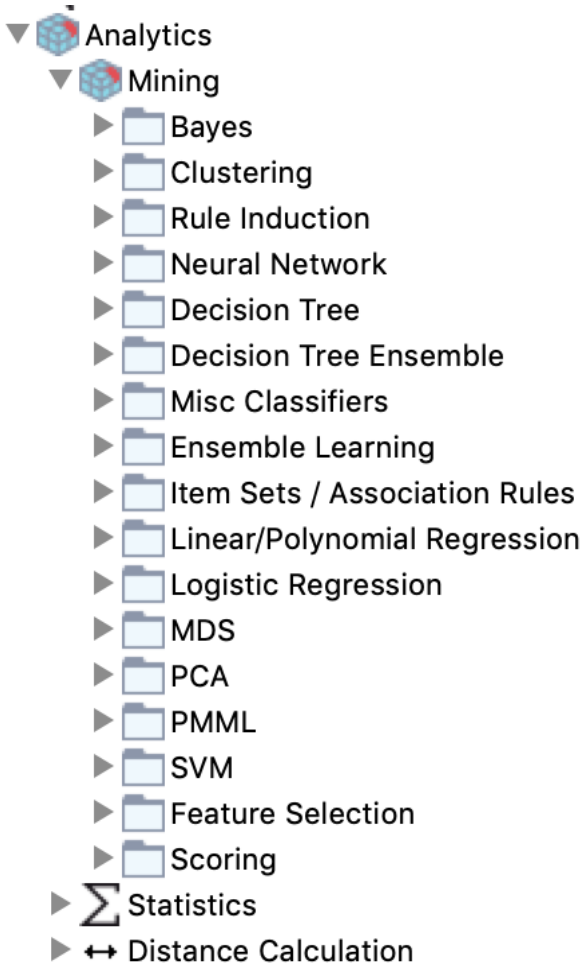


Рисунок 1.5 Засоби для аналізу програмного забезпечення KNIME

Великим недоліком є недостатня гнучкість налаштувань операторів, що зменшує ефективність даного програмного забезпечення.

1.3.4 Система WEKA

Система Waikato Environment for Knowledge Analysis (Weka) — це кроссплатформене програмне забезпечення для побудови сценаріїв аналітики великих даних, розроблене в Університеті Вайкато, Нова Зеландія. Мова програмування на який реалізована WEKA — Java.

Для роботи із даними у WEKA може працювати із наступними типами файлів представленими на рисунку 1.6:

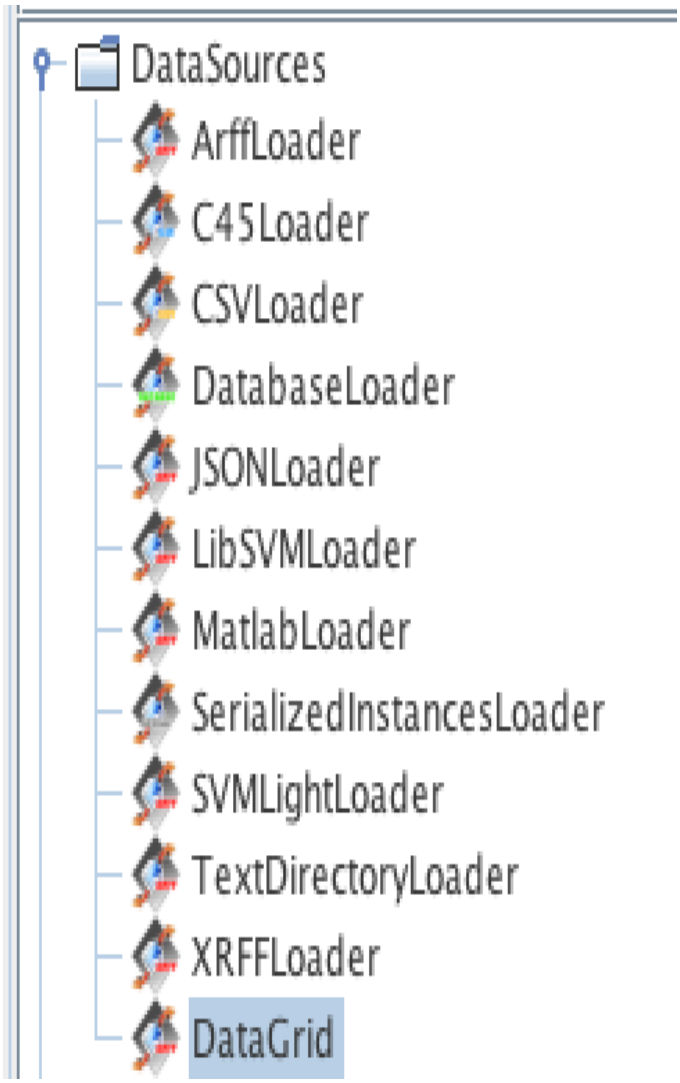


Рисунок 1.6 — Оператори для роботи із даними в WEKA

Фунціонал WEKA майже повністю досягає рівня, який задає Rapid Miner. Він забезпечує як і попередню обробку даних так і достатню кількість операторів проте не дуже зрозумілий інтерфейс, на який скаржаться у відгуках та розбиття операторів на не інтуїтивні категорії йдуть у протиріччя із основними вимогами до інструментальних засобів побудови сценаріїв аналітики.

1.3.5 Система Alteryx

Alteryx - американська компанія з комп'ютерного програмного забезпечення, що базується в Ірваїні, штат Каліфорнія. Для роботи із великими даними компанія пропонує чотири основні модулі — Connect, Promote, Server та Desinger. Саме останній із компонентів продукції компанії використовується для аналітики великих даних у простій та зручній візуальній формі за допомогою drag and drop редактора.

Програмне забезпечення надає оператори для роботи із різними типами структурованих файлів, як csv або excel, та бази даних Microsoft SQL Server та Oracle.

Для попередньої обробки даних в програмному забезпечені наявно дев’ятнадцять операторів, проте відсутня можливість для нормалізації вхідної інформації.

Серед алгоритмів та методів, які застосовані для аналізу великих даних були використані наступні:

- лінійна регресія;
- логістична регресія;
- наївний баєсів класифікатор;
- метод Random Forest;
- метод опорних векторів;

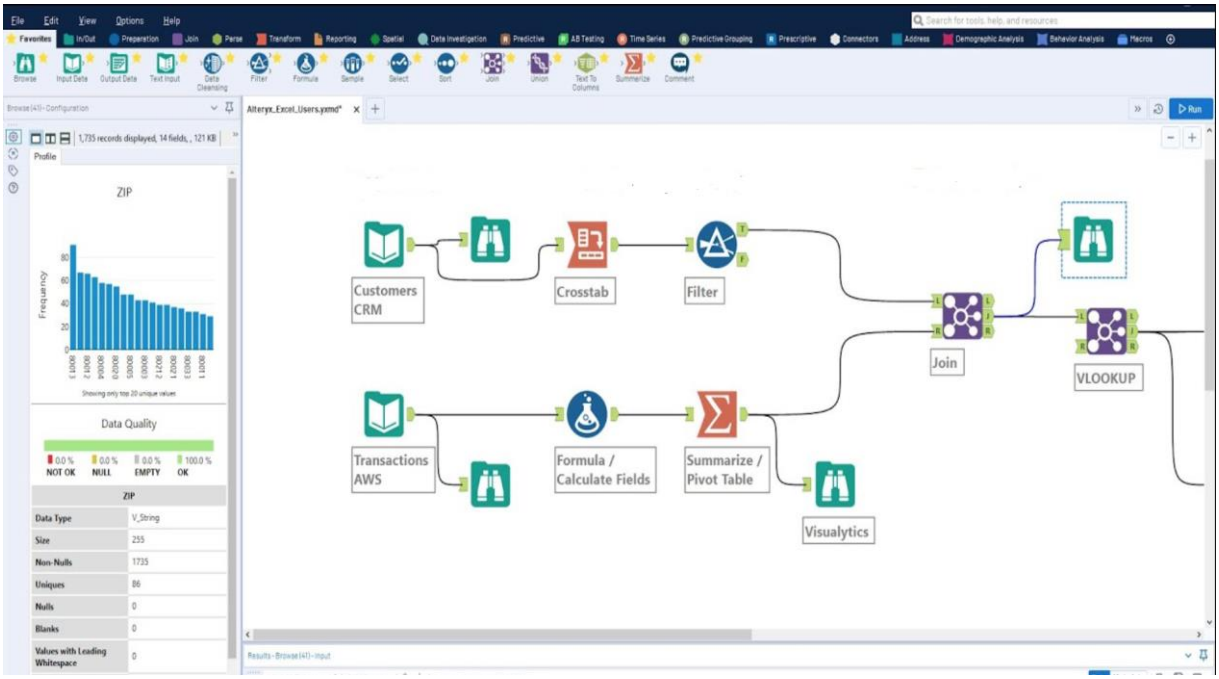


Рисунок 1.7 — Функціонал Alteryx Desinger

Для візуалізації у Alteryx Desinger використовується розвинута система генерації звітів, що дозволяє із широкого спектру заготовлених шаблонів візуалізувати інформацію багатьма способами та зберігати у різних форматів — від png до pdf..

Основним недоліком програмного забезпечення Alteryx Desinger можна назвати те, що воно не є кросплатформеним, а запускається лише на операційній системі Windows 10. Більш того, Alteryx Desinger — це проект із закритим вихідним кодом, з чого випливає що подивитися як реалізовані алгоритми для аналізу та обробки даних немає можливості. А отже неможливо сказати із якою ефективністю працює система.

1.4 Висновки до розділу 1

В даному розділі було розглянуто основні концепції та актуальність терміну великих даних. Із збільшенням інформації, яке почало генерувати людство за допомогою розвитку інформаційних технологій відбулися титанічні зміни у підходах зберігання та роботи із даними.

В ході розгляду напряму аналітики великих даних було виявлено, що він є одним із найперспективніших в сучасному світі, оскільки широко застосовуються як в науці для отримання проривних знань, так і в роботах підприємств різних розмірів для виділення скритих закономірностей та кореляцій, що можуть набагато збільшити потенційний дохід.

Було конкретизовано термін “сценарій аналітики” та представлені нові напрями аналізу даних, які стали можливі із застосуванням алгоритмів та методів обробки великих даних. Також було проаналізовані і виведені основні етапи з якими зустрічається бізнес аналітик при побудові сценаріїв.

В розділі був наведений перелік критеріїв, які необхідно враховувати при побудові інструментальних засобів аналітики сценаріїв великих даних та проведений аналіз існуючих програмних засобів. Були виділені основні переваги та недоліки.

2. РОЗРОБКА ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ПОБУДОВИ СЦЕНАРІЇВ АНАЛІТИКИ ВЕЛИКИХ ДАНИХ

2.1 Вибір мови програмування

Першим кроком, який необхідно зробити при розробки будь-якої системи – це обрати формат, у якому буде представлена кінцеве програмне забезпечення та стек технологій на якій воно буде написана.

Система інструментальних засобів побудови сценаріїв аналітики великих даних буде реалізована у вигляді кросплатформеного клієнт-серверного додатку для персональних комп'ютерів. Основні критерії, які необхідно оцінювати при виборі мови програмування є:

- мова повинна забезпечити простоту масштабованості системи у майбутньому. Тобто можливість легко та швидко додавати нові елементи до існуючої системи[13];
- мова повинна мати багатий та зручний інструментарій для роботи із великими даними. Особливо потрібно звертати увагу на засоби обробки так і візуалізації інформації;
- код програмного забезпечення конкретної мови повинен бути простим та зрозумілим;

З ростом популярності терміну великих даних та зацікавленість ринку у програмному забезпеченні для роботи із ним, все більше мов програмування почали отримувати свої технологій та фреймворки для роботи із ними. Почали навіть створюватись окремі мови спеціально для вирішення задач пов'язаних із big data та інтелектуального аналізу даних.

Найпопулярнішими мовами програмування із найбільш розвиненими технологіями на сьогоднішній день є:

- мова програмувааня Java;

- мова програмувааня Scala;
- мова програмувааня R;
- мова програмувааня Python.

2.1.1 Мова програмування Java

Мова Java один із лідерів рейтингів серед інших мов програмування, які застосовуються для роботи із великими даними. Мова Java є високорівневою строго типізованою С-подібною об’єктно-орієнтованою мовою, що широко застосовується в розробці сучасного програмного забезпечення.

Можливо це стало за допомогою технології Java Virtual Machine (JVM)[14], що дозволяє виконувати написаний байткод у код налаштований до специфікації будь-якій операційній системі та процесора, включаючи і мобільні пристрої на базах найпопулярніших операційних систем, таких як Android і iOS. Ще однією перевагою серед багаточисленних інших позитивних сторін Java можна виділити швидкість роботи із пам’яттю, що особливо важливо коли йде мова про роботу із надмірно великими масивами даних різних форм та об’єму.

Наступною перевагою Java постає той факт, що багато проектів для big data від фонду Apache Sowtware Foundation були написані саме на мові програмування Java. Основним для роботи із великими даними серед них можна назвати Hadoop.

Система Hadoop – це проект, що сформований із чотирьох основних модулів[15]. Центральним модулем для нього є Hadoop Common. Система, що відповідає за планування задач і управління кластерів YARN. Для пришвидшення роботи із даними використовується файлова система HDFS, а для обчислення використовується модуль Hadoop MapReduce. Оскільки два останні модулі є ключовими при роботі із великими даними на них потрібно розглянути детальніше.

Отже HDFS – це розподілена файлова система, основна перевага якої полягає в забезпеченні масштабованості і надійного зберігання великих об’ємів даних, яка досягається розбиттям великих кластерів інформації на стандартних серверах.[16] За структурою HDFS це ієрархічна файлова система, що складається із NameNode (вузол імені) який відповідає за роботу файлових операцій та DataNode (вузол даних), який

відповідає за самі операції над даними. Особливість архітектури полягає в тому, що вузол імені може бути лише один. Для роботи із файлами використовується принцип Write-once and read-many (одни раз записати – багато разів зчитати), що звільняє систему від класичної проблеми блокування типу “запис-читання”. Детальніше на архітектуру HDFS можна розглянути на рисунку 2.1.

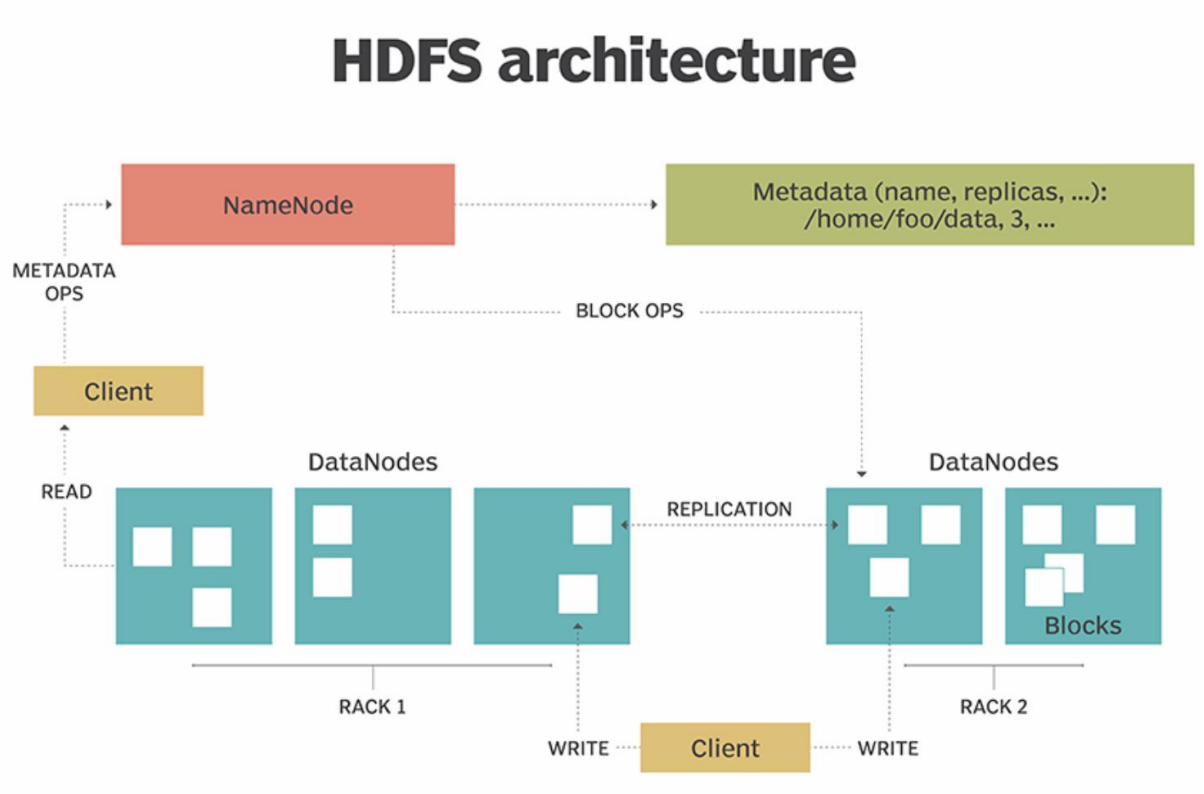


Рисунок 2.1 – Архітектура HDFS

Однак Java має і свої недоліки:

- код написаний на Java складний у довготривалій підтримці та займає набагато більше часу для написання та відладки;
- програє у інструментарію для візуалізації, який надають інші конкуренти, як Python або R;
- відсутність підтримки ітеративної розробки;
- стандарт REPL був доданий лише з 9 версії мови;

2.1.2 Мова програмування Scala

Scala, що походить від двох англійських слів “scalable” та “language”, які перекладаються як «масштабована мова»[17] — мова програмування високого рівня

з надійною системою статичного типу, що підтримує як об'єктно-орієнтоване так і функціональне програмування.

Мова Scala працює на Java Virtual Machine і має здатність безперебійно взаємодіяти із з Java. Тобто використовувати бібліотеки, викликати код, реалізовувати інтерфейси у Scala та навпаки. Однак є деякі функції та методи Scala, як розширені типи, до яких неможливо отримати доступ з Java. Виходячи з цього програмне забезпечення написане на Scala буде кросплафтормене.

Більш того за допомогою мови програмування Scala були написані наступні проектів для фонду Apache Sowtware Foundation для роботи із великими даними, такі як:

- проект Apache Spark;
- проект Apache Flink;
- проект Apache Kafka;
- проект Apache Samza[18];

Крім того, сам синтаксис мови програмування Scala є стислим. Кілька циклів можна замінити одним рядком, що робить його значно більш читабельним і швидким для написання великих програмних забезпечень, ніж стандартний Java. Це робить дозволяє коду на Scala бути оптимізованим і ефективним. Проте не типовість та відмінність від класичних С-подібних мов програмування стає і проблемою, оскілки для освоєння Scala і розуміння чужого коду може знадобитися набагато більше часу ніж у Java або Python.

Однією із проблем Scala можна виділити обмеженість сумісності нових версій. При кожному випуску нової версії із потрібним функціоналом оновлення системи в цілому може зайняти набагато більше часу. Порівняно із Python або R, Scala має не настільки багато бібліотек із додатковим функціоналом, що також збільшить час потрібний на розробку програмного забезпечення. Більш того великим обмеження при побудові системи може стати той факт, що компілятор Scala потребує потужні процесори для ефективного виконання коду.

2.1.3 Мова програмування R

Мова R — це спеціалізована мова програмування, а також одночасно програмне середовище для статистичних обчислень, аналізу та візуалізацію даних в графічному вигляді. Розробка та розвиток мови відбувався під впливом двох інших мов S та Scheme, від другої була успадкована більша частина семантики[19].

Мова R це універсальна мова, що підтримує багато парадигм програмування. Вона підтримує об'єктно-орієнтоване, імперативне, функціональне, процедурне та рефлексивне програмування. Мова R - це мова, яка використовує інтерпретатор, що дозволяє реалізовувати модульне програмування за допомогою функцій. Сценарій командного рядка в середовищі R дозволяє зберігати ряд складних етапів аналізу даних, що, спрощує повторне використання роботи з аналізу подібної інформації в майбутньому.

Крім того, R містить в собі функціонал для широкого спектру статистичних алгоритмів (включаючи лінійне та нелінійне моделювання, аналіз часових рядів, класифікацію, кластеризацію, згладжування тощо), а також широкий набір функцій для створення різного роду представлення даних. Ці функції можна легко розширити за допомогою пакетів. Окремим плюсом є те, що дистрибутиви R наявні на усіх популярних операційних системах, як Windows, Linux та MacOS.

Однією з особливостей програмування на мові R є вектори. Вектор — унікальна особливість, яка відрізняє R від більшості інших мов. Мова дозволяє застосовувати функції над векторами в одній операції без необхідності явного циклу foreach або while. Отже, складні операції можна виконувати над набором значень однією строчкою коду. Загалом, R асоціює атрибути до всіх структур даних, тому кожен вектор, список чи функція мають приховану карту, яка пов'язує символи зі значеннями.

Однією з сильних сторін R є численні додаткові пакети, які розповсюджуються у відкритому доступі. Пакети – це колекції функцій, набори даних, класи та додаткові засоби до стандартного набору функцій R. Ці пакети є аналогом бібліотекам Python. Наразі їх нараховується близько п’ятнадцяти тисяч[20]. Пакети включаються у себе додаткові можливості візуалізації, статистики, інтеграцію із іншими мовами

програмування, як Java, Python, C++. Встановлення бібліотек не є чимось складним і не потребує багато зусиль.

Однак у мови програмування R є свої недоліки. R це мова не загальне призначення і це означає, що код, написаний на R, не може бути розгорнутим у продакшені одразу. Його, як правило, треба підключити до програм написаних, наприклад, на Python або Java.

Мова R успадкувала від мови програмування S неможливість використанні динамічної або тривимірної графіки. Усі об'єкти R зберігає у локальну пам'ять, що може стати проблемою при роботі із великими масивами даних. Це призводить до того, що швидкість роботи програмного забезпечення значно нижче ніж аналогічні рішення написані на інших мовах програмування. Іншою проблемою мови відмінний від класичних C-подібних мов програмування, що ускладнює не тільки написання коду та відладку, а також підтримку системи у майбутньому.

2.1.4 Мова програмування Python

Одним із найпопулярніших виборів серед мов програмування для роботи із великими даними є Python. За опитуваннями сайту https://towardsdatascience.com проведеного у 2019 році серед працівників сфери аналітиків великих даних ця мова програмування є лідером. Детальніше чарт представлений на рисунку 2.3.

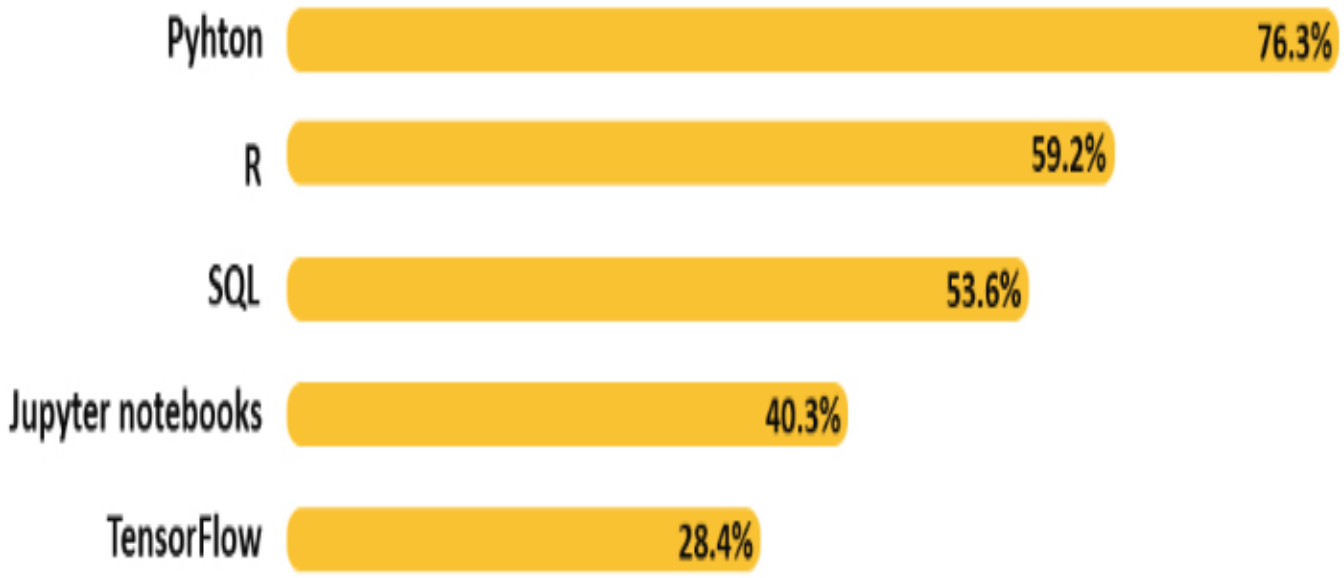


Рисунок 2.3 — Популярність технологій для аналітики великих

Такий ріст популярності мови Python можна зумовити високу швидкість опанування мови та простоту написання програмного коду. Поєднання вбудованих структур даних високого рівня та динамічного типізації робить програми Python в 3–5 разів коротшими, ніж еквівалентні програми на Java. Чіткий синтаксис і читабельністю, яка досягається через обов'язковість відступів висловлювань полегшує читання коду.

Ще однією великою превагою мови Python є можливість працювати через API із найбільш популярними технологіями та фреймоварками для роботи із великими даними та машинним навчанням, які початково не використовували Python. Серед них можна виділити і згадані вище проекти від Apache.

Наступний плюс Python, як мови програмування, буде велика кількість бібліотек, для аналізу та обробки великих даних[21]. Серед них можна виділити scikit-learn та pyspark.

Оскільки Python – це мова загального призначення, то існує велика кількість засобів та фрейм ворків, що повністю забезпечує можливість створювати різне програмне забезпечення від настільних додатків (використовуючи для побудови інтерфейсу засоби PyQt або GTK+) до веб-сервісів (Django або Flask). Python підтримує можливість працювати, як із реляційними базами даних серед яких Mictosoft SQL Server, MySql Server, Orcarle, Postgresi і інші, так й NoSQL, такими як Mongo.

Не дивлячись на той факт, що мова програмування R має кращий інструментарій для візуалізації даних, Python має багато бібліотек що створює непогану конкуренцію функціоналу R. Серед найбільш відомих можна назвати наступні рішення:

- бібліотекаMatplotlib;
- бібліотека Ploty;
- бібіліотека Ggplot;
- бібліотека NetworkX.

Серед основних недоліків Python, можна зазначити його швидкість. Оскільки код програми виконується інтерпретатором рядок за рядком, а не як в компільованих

програмах все одночасно це сильно знижує ефективність кінцевого програмного забезпечення.

Наступна проблема теж витікає із використання інтерпретатора. Помилку, яка була допущена при написанні програмного коду неможливо відслідкувати під час запуску програми оскільки не відбувається компіляції коду. Більш того мова програмування Python використовує динамічно-типізовані змінні, що робить тестування більш важким і хаотичним. Більшість помилок будуть виявлятися лише під час виконання.

Структура дизайну Python така, що вона використовує великі об’єми оперативної пам'яті під час обробки в порівнянні з іншими мовами, як C, C ++, C# або Java. Це робить мову програмування Python повним не найкращим вибором для розробки програмного забезпечення в умовах, коли присутні жорсткі обмеження на пам’ять.

2.1.5 Висновки порівняльного аналізу мов програмування

Результати аналізу, щодо вибори технологій та мов програмування для побудови системи інструментальних засобів аналітики великих даних можна звести до таблиці 2.1.

Таблиця 2.1 – Порівняння характеристик різних мов програмування

	Java	**Scala**	**R**	**Python**
Швидкість	+	+		
Легкість опанування				+
Зручність для аналітики даних		+	+	+
Підтримка big data	+	+	+	+
Взаємодія з іншими мовами програмування			+	+

Таблиця 2.1 – Порівняння характеристик різних мов програмування(продовження)

Мова програмування широкого застосунку	+			+

Не дивлячись на те, що Python доволі сильно програє Java у швидкості виконання, по всім іншим пунктам ця мова програмування є лідером у сфері роботи big data. Високо розвинута інфраструктура для аналітики та роботи із великими даними, різноманіття бібліотек для візуалізації інформації, можливість створювати як і графічний інтерфейс, так і серверну частину програмного забезпечення, простий для написання та підтримки код дозволяє швидко розробити необхідну систему.

2.2 Вибір бібліотеки мови програмування Python для побудови сценаріїв аналітики великих даних

Оскільки на мові програмування Python можна повністю побудувати систему для роботи із великими даними, необхідно обрати бібліотеки та фреймворки як для побудови графічного інтерфейсу, так і для обробки інформації.

Для того щоб підвищити ефективність виконання математичних функцій над наборами даних у системі реалізованій на Python буде використовуватися бібліотека NumPy та SciPy. Бібліотека NumPy це модуль із відкритим кодом, який надає загальні математичні і числові операції у вигляді пре-скомпільовані, швидких функцій. Вони об'єднуються в високо рівневі пакети та забезпечують функціонал, який можна порівняти з функціоналом MatLab не програючи при цьому у швидкості.[22] Бібліотека NumPy (Numeric Python) надає базові методи для маніпуляції з великими масивами і матрицями. SciPy (Scientific Python) розширює функціонал numpy величезною колекцією корисних алгоритмів, таких як мінімізація, перетворення Фур'є, регресія, і інші прикладні математичні техніки[23]. Головною особливістю numpy є масиви. Вони схожі зі стандартними списками в Python, виключаючи той

факт, що елементи масиву повинні мати однаковий тип даних, як float і int. З масивами можна проводити числові операції з великим обсягом інформації в рази швидше і, головне, набагато ефективніше ніж зі списками.

Для базового аналізу даних буде використана бібліотека pandas, що отримала назву від економетричного терміну для багатовимірних структурованих наборів даних[24]. Завдяки тому, що найкритичніші частини реалізовані на мові програмування C, pandas є дуже оптимізованою та високошвидкісною бібліотекою. Данна бібліотека представляє числові значення у вигляді структур даних ndarray NumPy і зберігає їх в безперервних блоках пам'яті. Ця модель зберігання даних дозволяє економно витрачати пам'ять і швидко отримувати доступ до значень. Основною одиницею є об’єкт типу DataFrame, що підтримує вбудовану індексацію. За допомогою pandas можна вирішувати наступні задачі:

- фільтрації даних;
- злиття різних наборів даних в один за різними критеріями;
- маніпуляції із типами даних;
- очистка даних;
- можливість зчитування із різних форматів даних;
- підтримка візуалізації;
- отримання зрізів.

Для розвинутого аналізу даних буде використовуватися бібліотека scikit-learn. Це універсальна бібліотека з відкритим вихідним кодом для інтелектуальної обробки інформації[25]. Одне з основних переваг бібліотеки полягає в тому, що вона працює на основі декількох поширених математичних бібліотек, і легко інтегрує їх один з одним, як наприклад вище згаданий NumPy. Ще однією перевагою є широка спільнота і докладна документація. Бібліотека scikit-learn широко використовується для промислових систем, в яких застосовуються алгоритми класичного машинного навчання для аналізу великих даних. За допомогою бібліотеки можна вирішити наступні задачі:

- задачі классифікації;
- задачі регресії;

- задачі кластеризації;
- нормалізації даних;
- очищення даних;
- задачі попередньої обробки даних.

Для реалізації графічного інтерфейсу буде використовуватися PyQt. Фрейм ворк PyQt - це прив'язка Python для Qt, що представляє собою набір бібліотек C ++ та інструментів розробки, які включають незалежні від платформи абстракції для графічних інтерфейсів користувача (GUI), а також функціонал для роботи із мережами, потоками, регулярними виразами, базами даних SQL, SVG, OpenGL, XML та багато іншого[26]. Актуальна версія фрейм ворку п’ята, що представляє собою більше ніж п’яти сотень різноманітних класів та більше шести тисяч функцій та методів. Ще однією фрейм ворку є його кросплатформеність.

2.3 Формування фунціоналу інструментальних засобів побудови сценаріїв аналітики великих даних

Якщо розглядати побудову сценарії, як певну послідовність кроків, які необхідно виконати для ведення аналітичної діяльності, то необхідно чітко розмежувати категорії, до яких буде належати кожен крок. В ході підготовчих етапів побудови інструментальних засобів аналітики було виведено наступні п’ять груп, на які були розбиті усі функціональні одиниці сценарію, що представлені на рисунку 2.3.

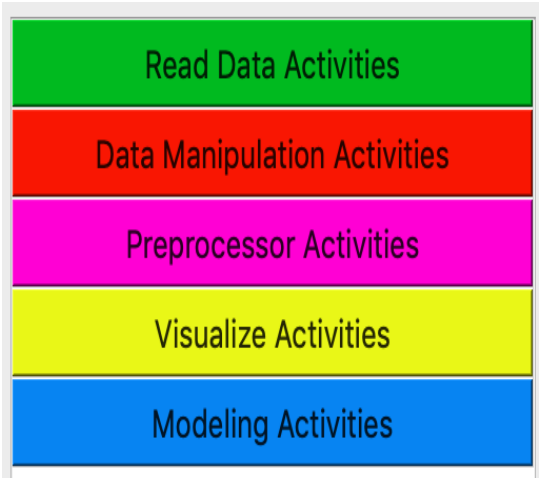


Рисунок 2.3 — Основні категорії доступних операцій в системі

В реалізованій системі кожна окрема одиниця сценарію називається віджет і цей термін буде використовуватися далі по тексту.

2.3.1 Віджети для зчитування даних

Перше, що необхідно зробити аналітику це завантажити дані із стороннього джерела. Оскільки інформація може зберігатися у різних форматах та буті експортована із різних систем, інструментальні засоби повинні надавати широкий спектр можливостей для можливості підключення до різних типів файлів.

Найпоширенішим та найпростішим способом збереження даних є самий простий текстовий файл із розширенням .txt. Кожна строчка файлу повинна закінчуватися на символ переносу на новий рядок, а знак-сепаратор що відокремлює окремий набір інформації повинен бути однаковим. Реалізацію віджета для роботи із текстовим файлом в системі можна побачити на рисунку 2.4.

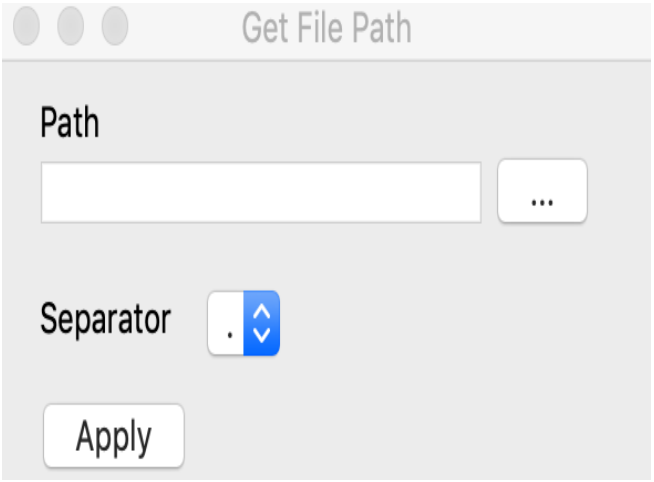


Рисунок 2.4 — Віджет для роботи із текстовим файлом

Одним із найпопулярніших шляхів зберігання інформації на будь-якому підприємстві є Excel файл. Програмного забезпечення від Microsoft представляє інформацію у добре структурованому виді використовуючи сітку комірок, розташованих у пронумерованих рядках та стовпчиках з літерними назвами[27]. Більш того один файл може мати необмежену кількість листів із подібними наборами даних. Експортувати дані із системи у вигляді excel файлу має можливість будь-яка CRM система. Реалізацію віджета представлено на рисунку 2.5.

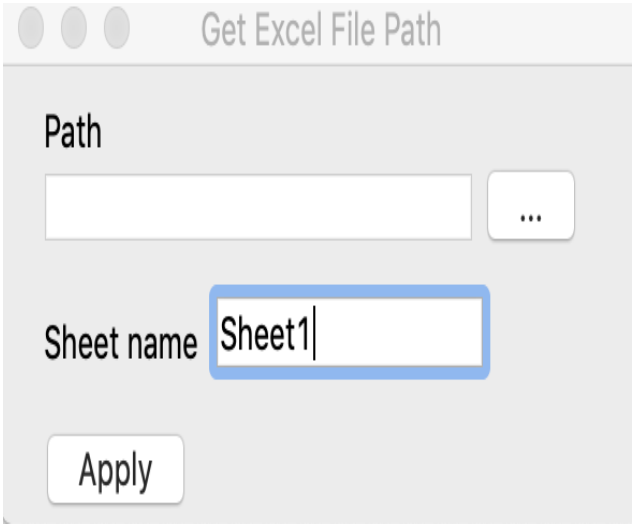


Рисунок 2.5 — Віджет для роботи із файлами Excel

Використовуючи бібліотеку pandas система інструментальних засобів побудови сценаріїв аналітики дозволяє завантажити дані із будь-якого файлу Excel, правильно вказавши при цьому назву листа. Єдина вимога — файл не повинен містити графіки або бути розбитий на під таблиці.

Із розвитком інтернету та необхідністю постійної комунікації між клієнтом та сервером відбувся стрімкий розвиток формату JSON. Це абревіатура від JavaScript Object Notation, хоча зараз об’єкти такого типу доступні і в інших мовах програмування[28]. Компактний JSON являє собою гарну альтернативу XML і вимагає куди менше форматування контенту. Об'єкт JSON це формат даних - ключ-значення, який зазвичай відображається в фігурних дужках. Ключі в JSON знаходяться з лівого боку від двокрапки. Їх потрібно обертати в дужки і це може бути будь-який рядок. Головна вимога — ключі повинні бути унікальними. JSON значення знаходяться з правого боку від двокрапки. Значення може бути одним з шести типів даних: рядком, числом, об'єктом, масивом, приймати значенням True/False або null. Вимогами до файлів існують наступні:

- файл повинен бути із розширенням .json;
- бути закодованим у форматі UTF-8;
- структура файла повин бути простою.

Реалізацію віджету для роботи із json файлами можна побачити на рисунку 2.7.

Рисунок 2.7 — Віджет роботи із json файлами

Кожне підприємство застосовує в своїй роботі реляційні бази даних для збереження накопиченої інформації. А отже можливість завантажувати звідти накопичені дані для подальшого аналізу полегшлено б життя аналітика. Більш того спеціаліст, який має досвід із мовою SQL, за допомогою запитів може зменшити підготовчий етап із даними до мінімуму написавши потрібний join або group by запит одразу до декількох таблиць. Тому необхідно надати вибір до якої саме реляційною бази буде підключатися аналітик, оскільки для цього необхідно використовувати спеціальний драйвер. При створені системи було обрано надати можливість підключатися до Microsoft SQL Server, MySQL Server та PostgreSQl Server. Реалізацію віджету для роботи із реляційними базами даних представлено на рисунку 2.6.

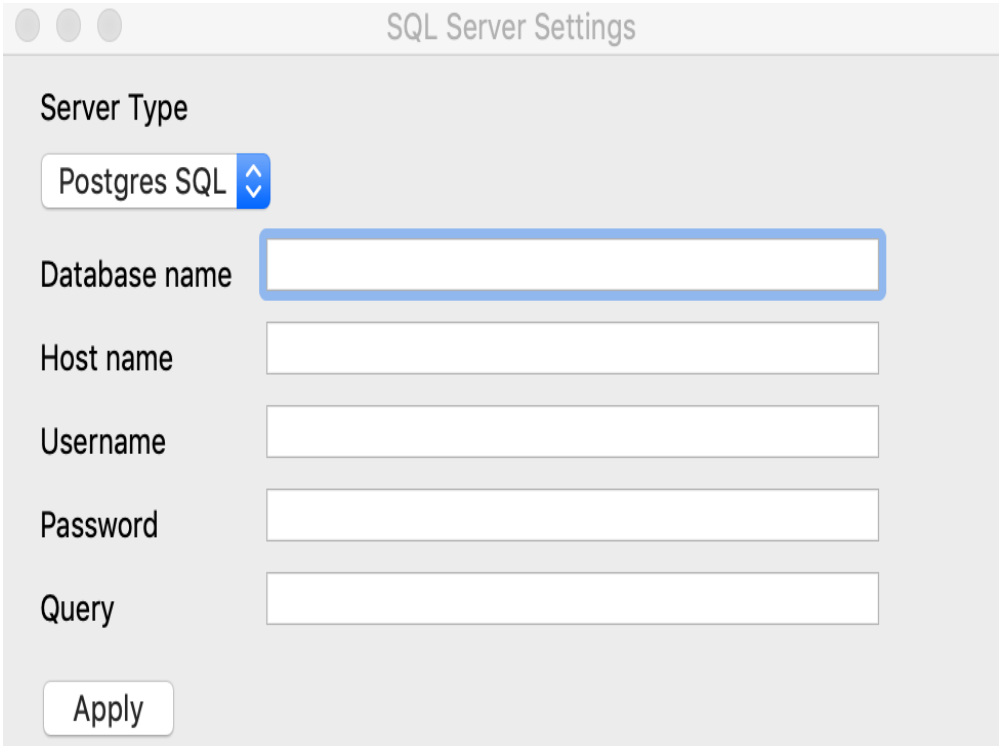


Рисунок 2.6 — Віджет роботи із реляційними базами даних

Ще однин віджет для роботи з даними, який був реалізований в системі націлений на роботу із файлами XML[29]. Абревіатура походить Extensible Markup Language, з особливим акцентом на markup (розмітка). Людина можете створювати текст і розмічати його за допомогою тегів, перетворюючи кожне слово, пропозицію або фрагмент в ідентифіковану, сортовану інформацію. Створювані файли, або екземпляри документа, складаються з елементів (тегів) і тексту. Чим більше описових елементів, тим більше частин документа можна ідентифікувати при обробці файлу. У XML можна створювати свої власні елементи, що дозволяє точно представляти фрагменти даних. Документи можна не просто розділяти на абзаци і заголовки, а й виділяти будь-які фрагменти всередині документа. Щоб це було ефективно, потрібно визначити кінцевий перелік своїх елементів і дотримуватися його. Елементи можна визначати в Описі типу документа (Document Type Definition - DTD). Для роботи в системі файл повинен бути закодований в форматі UTF-8.

Наступний формат зберігання інформації для реалізації в системі було обрано формат csv. В файлах такого типу міститься інформація, а поля розділяються комою, оскільки абревіатура означає comma separated values. Незважаючи на назву дуже часто широко використовуються і інші роздільники, тобто формат CSV дуже часто плутають із більш широким поняттям delimiter separated values (DSV) [30]. Серед них можна зустріти і крапку з комою, і вертикальні лінії, і табуляцію. Основна проблема формату — він не стандартизований в повному обсязі. Ідея використовувати сепаратору для розділення очевидна, але при такому підході виникають проблеми якщо вихідні табличні дані містять ці роздільники або більше, наявні переведення рядків на нову. Можливим рішення таких ситуацій стає обрамлення даних в лапки, проте виникає проблеми, коли строки вхідних даних теж містять лапки. З цього треба накласти обмеження на файли, які можна застосовувати в системі:

- комірки, що містять строкові значення із знаком роздільника повинні бути взяті у лапки;
- файл повинен бути у форматі csv та використовувати формат UTF-8.

Реалізацію віджета роботи із csv файлом представлені на рисунку 2.8.

Рисунок 2.8 — Віджет роботи із csv файлами

2.3.2 Віджети для маніпуляції із даними

Оскільки дані можуть бути отримані із різних джерел тому не завжди можливо провести необхідні підготовчі дії до завантаження інформації в систему інструментальних засобів моделювання сценаріїв аналітики. Отже необхідно надати такий функціонал кінцевому користувачу.

По-перше у аналітика повинна бути можливість об’єднати два однакових за структурою даних, які були завантажені із двох різних джерел даних. Наприклад, інформація про закупівлі із різних філій розподіленого підприємства. Щоб досягнути такого результату можна використати метод concat бібліотеки pandas[31]. Інтерфейс віджету Concatenate, що надає такий функціонал представлено на рисунку 2.9.

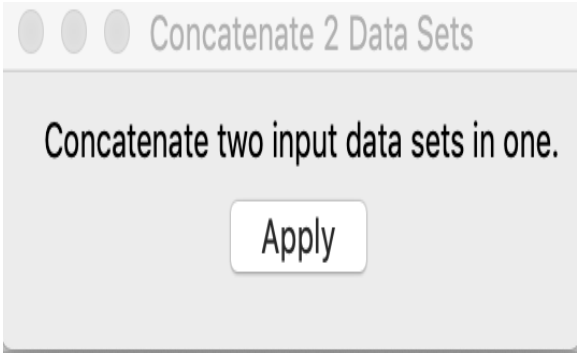


Рисунок 2.9 — Віджет об’єднання двох наборів даних в один

Не менш важливим є фунціонал для злиття двох наборів даних в один ґрунтуючись на певній однаковій колонці за вибором кінцевого користувача. Для цього знов таки можна використати функціонал бібліотеки pandas. Проте необхідно надати вибір, що робити при наявності пустих колонок у другому дата сеті. Для цього в системі реалізовані дві можливі функції:

- задати відсутнім параметрам значення NaN і отримати на виході набір даних, такого ж розміру, що і на вході;

- прибрати строки, що не мають співпадіння і отримати на виході набір даних меншого розміру.

Віджет для злиття двох джерел інформації представлений на рисунку 2.10.

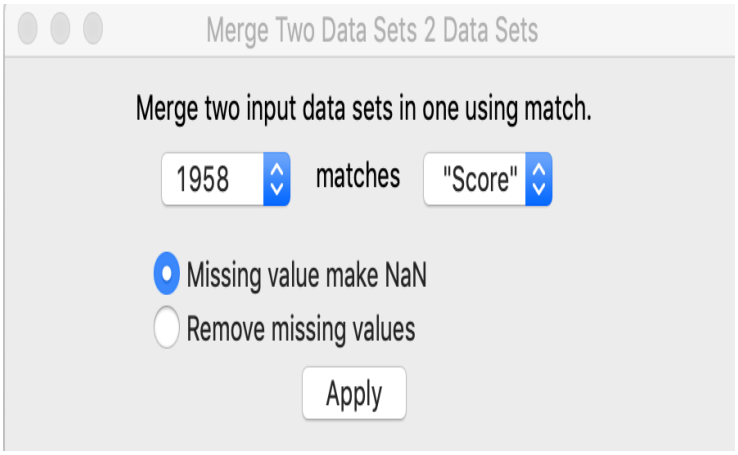


Рисунок 2.10 — Віджет злиття двох наборів даних в один

Оскільки не завжди набори даних можуть мати лише релевантну інформацію для подальшого аналізу, необхідно надати фунціонал, який би реалізовував би можливість наклади обмеження на завантажені дані. Для цього в системі було реалізовано віджет, що задає правила за якими відбувається фільтрація даних. Кожне правило виконується послідовно крок за кроком до одного дата сету, тому необхідно враховувати порядок їх задання. Реалізацію віджету для такого фільтрації зображено на рисунку 2.11.

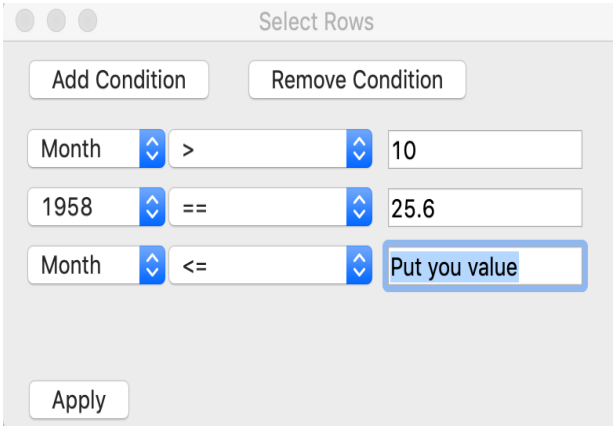
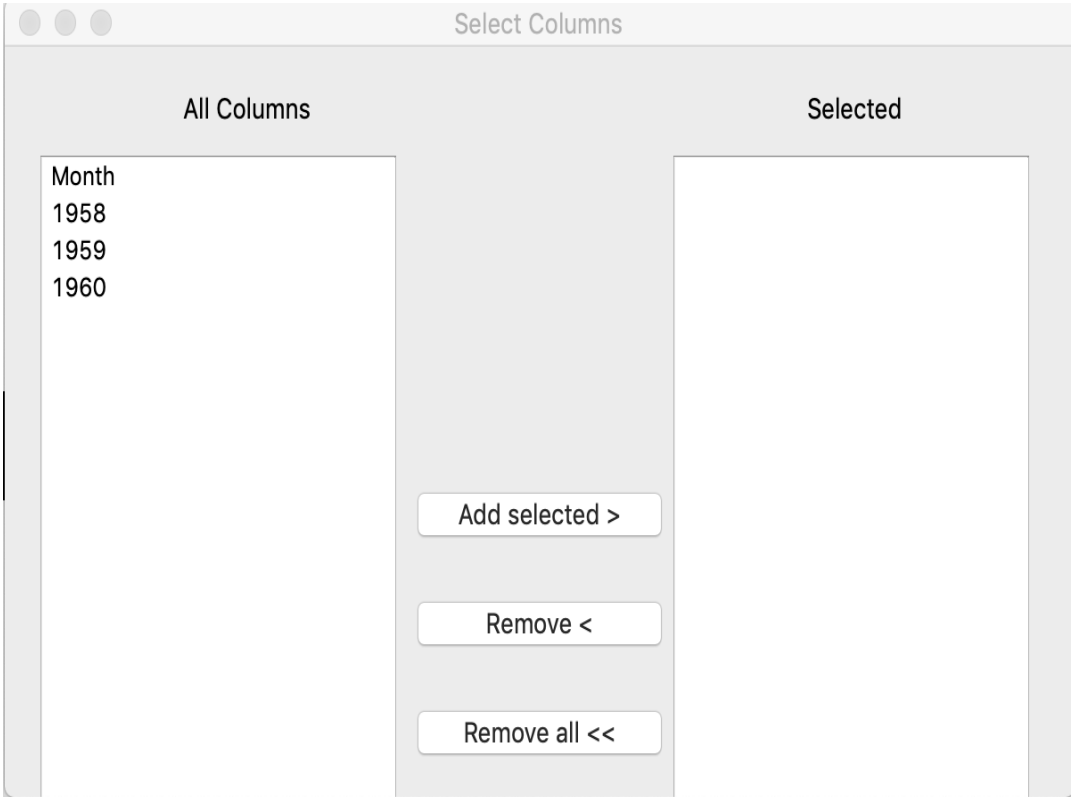


Рисунок 2.11 — Віджет для задання правил фільтрації вхідних даних

Більш того не усі колонки є важливими для роботи із даними і щоб не використовувати більше необхідної пам’яті необхідно надати можливість аналітику обирати, які саме стовпчики використовувати в ході аналізу. Приклад такого віджету представлено на рисунку 2.12.



2.12 Віджет для фільтрації даних за колонками

Інформація, яку використовує аналітик не завжди є добре структурованою та відображає усю повноту предметної області. Так, наприклад, в таблиці буде представлена колонка із заробітною ставкою співробітника за зміну та кількість відпрацьованих за певний період часу робочих днів. Проте аналітику необхідно мати саме точну суму, яку заробили працівники певного відділу. Саме для цього в системі було реалізовано віждет для створення власних колонок на основі математичних формул, які підтримують синтаксис стандартної бібліотеки мови python. Представлення віджету для побудови власної колонки представлено на рисунку 2.13.

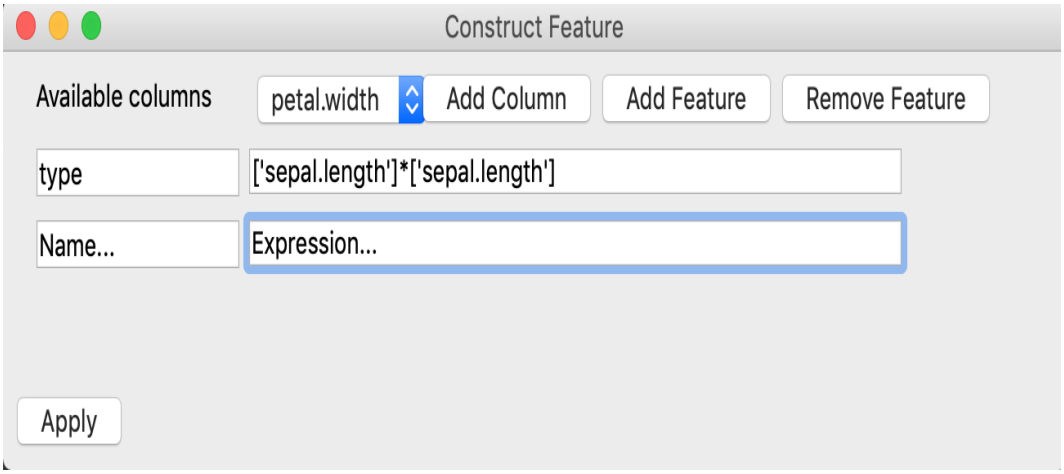


Рисунок 2.14 — Віджет для побудови власних колонок

Також обов’язково аналітик повинен при необхідності готувати дані для наступного навчання моделі в алгоритмах роботи із великими даними. Для цього він повинен мати можливість розбивати інформацію на певні набори і перевіряти їх доцільність. Один із способів — обирати у випадкові, не пов’язані між собою колонки для пошуку скритих кореляцій, які неможливо помітити на перший погляд. Для цього у системі представлений віджет, що дозволяє задати певну кількість необхідних колонок і граючись із ймовірнісним алгоритмом мови Python отримувати різні набори.

Гостро стає питання навчання моделі при обмеженій кількості наявній інформації. Саме для цього зазвичай дані, які використовуються для аналізу, необхідно розбити на два набори. Один дата сет буде використаний для отримання коефіцієнтів обраного алгоритму, другий — застосований для перевірки ефективності отриманої моделі. Стандартна практика розбити таблицю на у співвідношенні сімдесят на тридцять[32].

Існує також необхідність розбивати набори даних на декілька підмножин у випадку, коли неможливо це зробити із файлом поза системи. Наприклад, якщо джерелом даних слугує JSON або XML файл, який не був відформатований у зручному вигляді для людини. Віджет в системі дозволяє використовуючи методи бібліотеки pandas отримати набори без повторень. Результат реалізації представлений на рисунку 2.15.

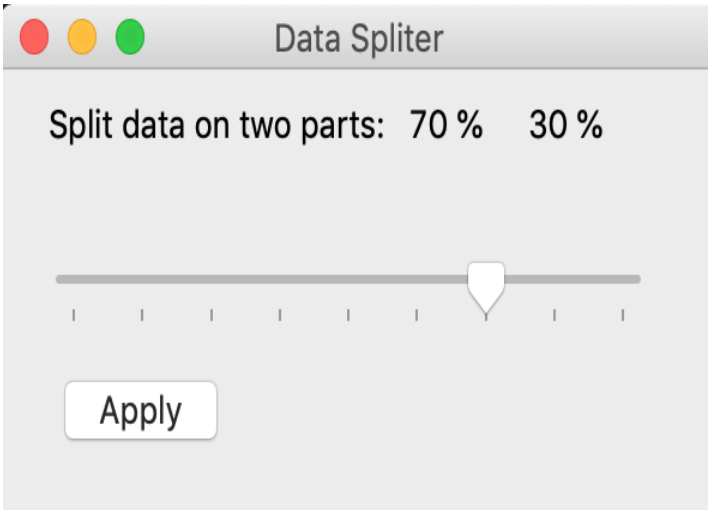


Рисунок 2.15 — Віджет для розбиття набору даних на два у заданому користувачем відношенні

Отримані результати в ході виконання сценарію аналітики великих даних можна зберігати у файлі формату csv, що буде використовувати кому як роздільник. Для цього необхідно буде обрати директорію на будь-якому диску і задати назву файлу.

2.3.3 Віджети для попередньої обробки даних

Перш ніж розпочати виконання обраних методів та алгоритмів інтелектуального аналізу даних, зазвичай першочерговим стає завдання розгляду та оцінки завантажених наборів інформації. Оскільки чим більша помилка наявна у вхідних даних, тим менше представляє цінність та коректність вихідних результатів. Тому ця проблема стає першочерговою для будь-якого аналітика, а з цього виникає необхідність реалізувати в інструментальних засобах для побудові сценаріїв функціонал для попередньої обробки даних.

Кожен дата сет різний і може мати унікальні проблеми однієї із основних категорій[33]:

- відсутність даних;
- наявність дублікатів;
- невідповідність даних.

Зазвичай такі помилки виникають через велику кількість факторів від людської помилки при введені інформації у системі, наявності рядків дублікатів до банальних проблем із приладами, які використовуються для накопичення даних та проблем із накопичувальними пристроями.

Одним із можливих засобів для вирішення цієї проблеми — відкидання рядків, що мають значення null у будь-якій із колонок. Це найпростіший спосіб дозволяє мати у дата сеті лише заповнені рядки. Проте це може значно зменшити об’єм дата сету, тому ще одним способом вирішення проблеми є заповнення пропущених елементів середнім значенням із усіх наявних в певній колонці значень[34].

Оскільки для алгоритмів аналізу великих даних мають значення лише цифрові параметри то необхідно також реалізувати функціонал, що дозволить одразу видаляти текстові значення із вхідного набору даних.

Більш того алгоритми для інтелектуального аналізу інформації працюють краще та швидше зближуються, коли функції знаходяться у порівняно схожих масштабах або близьких до нормально розподілення. Найкраще для цього підходить алгоритм із бібліотеки sklearn — MinMaxScaler[35]. Віджет для нормалізації необхідних колонок у діапазоні від 0 до 1 представлено на рисунку 2.16.

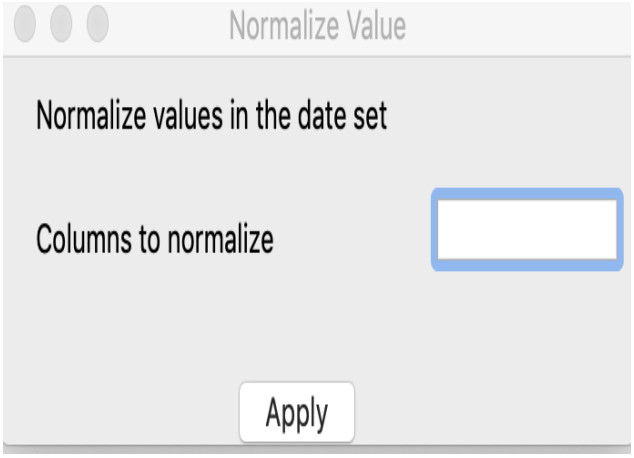


Рисунок 2.16 — Віджет для нормалізації колонок обраних користувачем

2.3.4 Віджети для візуалізації

Для візуалізації отриманих результатів було обрані та реалізовані наступні графіки:

- лінійний графік залежності;
- точкова діаграма;
- стовпчикова діаграма;

Вікно налаштування віджету, що дозволяє обрати ось X та ось Y із наявних в дата сеті колонках, представлене на рисунку 2.17.

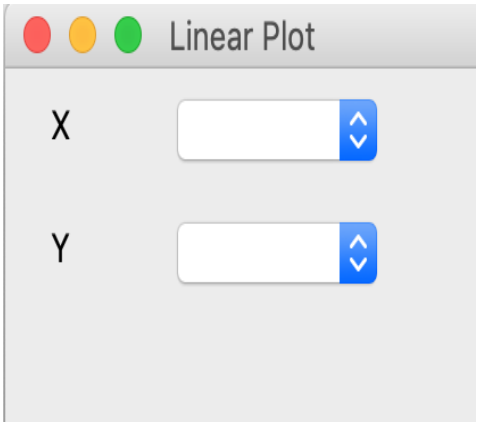


Рисунок 2.17 — Віджет налаштування осей для графіку лінійної залежності

Візуалізація в системі реалізована за допомогою бібліотеки Matplotlib Python для двовимірних графіків. Крім широкого спектру графіків та засобів маніпуляції із ним, як масштабування, бібліотека надає можливість зберігати рисунок у форматі png.[36] Приклад графіка точкової діагрмаи побудованого за допомогою засобів Maplotlib представлено на рисунку 2.18.

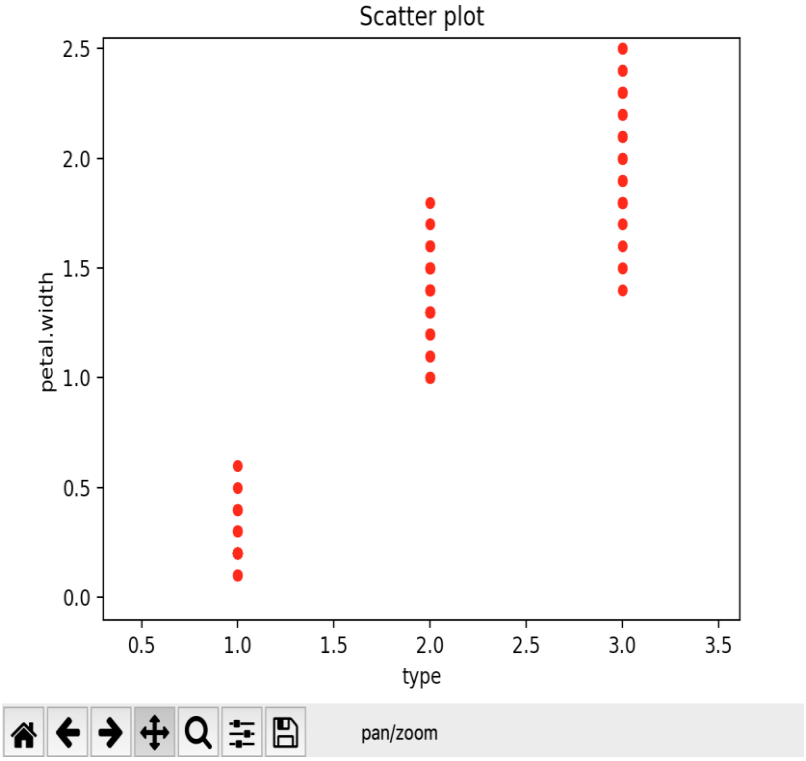


Рисунок 2.18 — Точкова діаграма залежності ширина пелюстки від типу

2.3.5 Віджети для інтелектуального аналізу даних

Найважливішим етапом для роботи аналітика стає вибір методів та алгоритмів, що будуть використані для отримання результатів. С розвитком великих даних був сформований новий напрям для обробки інформації, що називається data mining[37]. Така назва була отримана завдяки маркетинговим цілям[38], хоча переклад в українській літературі, а саме інтелектуальний аналіз даних, більш точно відображає сенс цього процесу. За суттю це автоматизований пошук даних, заснований на аналізі величезних масивів інформації. За кінцеву мету береться ідентифікація тенденцій і паттернів, що було неможливо виділити та класифікувати при традиційних підходах до аналізу. Для сегментації даних і оцінки ймовірнісних коефіцієнтів подальших подій використовуються складні математичні функції.

Основна перевага інтелектуального аналізу даних над класичними полягає у підходах в роботі із моделлю. По-перше, статистичні методи вимагають постійного контролю аналітиків для коригування та підтвердження справності роботи моделі аналізу, що робить їх практично не автоматизованими та не стійкими до будь-яких змін. По-друге, традиційні методи зазвичай працюють з невеликими обсягами інформації, так званими вибірками, що істотно спотворює прогнозованість результатів. Алгоритми data mining можуть націлені на роботу із великими даними.

Процес інтелектуального аналізу даних використовуються для вирішення наступних типів задач[39-41]:

- виявлення аномалій — пошук незвичних записів даних у системі, які можуть бути цікавими і потребують подальшого дослідження для розуміння навколишнього середовища або помилкових наборів. Аномалії можуть бути викликані непередбачуваними зовнішніми або внутрішніми факторами;
- навчання правилам асоціації (моделювання залежності). Під цим процесом розуміється побудова правил залежності між змінними у наборі даних. Наприклад, магазин або сервіс може збирати дані про звички користувачів. Це іноді називають аналізом ринкових кошиків;
- кластеризація даних — це завдання, що полягає у виявленні групи та структури в даних, які мають певні суміжні характеристики. Але основна особливість при цьому аналітик не повинен визначати структури у даних до проведення аналізу, вона формується вході виконання алгоритму;
- класифікація даних — це завдання для узагальнення відомої структури і застосовувати її для приналежності нових даних до сформованих категорій;
- регресія даних — це задача, ціль якої сформувати функцію, яка буде моделювати дані з найменшою помилкою, тобто для оцінки зав’язків між даними для подальшого передбачення результатів на нових вхідних даних;
- узагальнення результатів — це забезпечення більш компактного подання набору даних, включаючи створення візуалізації у різних форматах та генерація звітів.

При побудові інструментальних засобів для побудови сценаріїв аналітики великих даних були використані наступні алгоритми для інтелектуального аналізу даних із бібліотеки scikit-learn:

- лінійна регресія;
- логістична регресія;
- наївний баєсів класифікатор;
- варіації методу опорних векторів;
- алгоритми Random Forest для регресії;
- алгоритми Random Forest для класифікації.

Лінійна регресія — лінійний підхід до моделювання взаємозв'язку між скалярною величиною, що зазвичай називають залежною змінною від іншого вектора, який може бути як одно, так і багатомірним, який прийнято характеризувати як незалежну змінну. Зв'язки моделюються за допомогою лінійних функцій предиктора, невідомі параметри моделі яких вираховуються за допомогою незалежних змін. Отримані вході роботи моделі називаються лінійними.[42] Як і всі форми регресійного аналізу, лінійна регресія зосереджується на умовному розподілі ймовірностей відповіді з урахуванням значень предикатора, а не на спільному розподілі ймовірностей усіх цих змінних, що є областю багатовимірного аналізу. Використовують отримані моделі для:

- прогнозування;
- передбаченні;
- зменшення помилок;
- пояснення залежності.

Використовуючи методи, які надає бібліотека scikit-learn користувачу надається можливість обрати залежну змінну, обрати вектор незалежних змін, зберігати коефіцієнти моделі, а також отримувати результат коефіцієнту детермінації, що буде вказувати на ефективність моделі. Чим ближче результат до значення 1, тим краще було проведене навчання. Результат реалізації віджета можна побачити на рисунку 2.19.

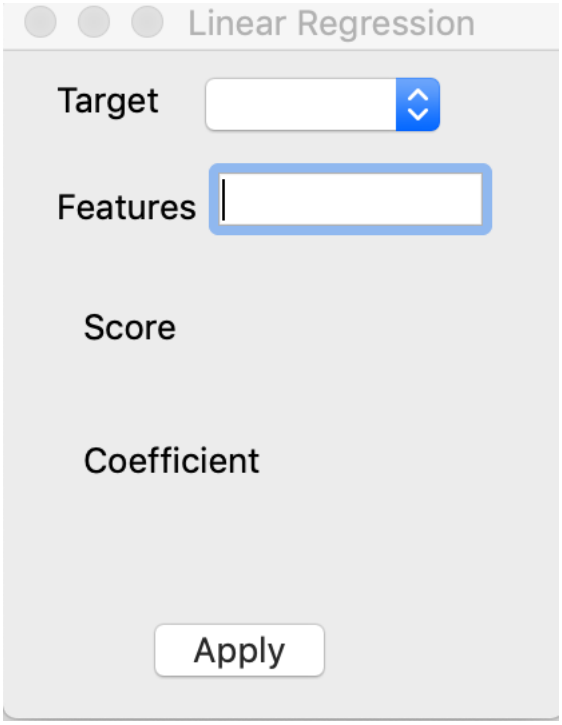


Рисунок 2.19 — Віджет алгоритму лінійної регресії

Наступний алгоритм доданий в систему – це логістична регресія. В цілому логістична регресія це різновид множинної регресії, загальне призначення якої полягає в аналізі зв'язку між декількома незалежними змінними і залежною змінною[43]. Вона застосовується, якщо залежна змінна є може приймати лише одне із двох значень (правда/брехня, нуль/один, тощо). Логістична регресія може бути двочленною, порядковою або багаточленною. Біноміальна або бінарна логістична регресія стосується ситуацій, коли спостережуваний результат для залежної змінної може мати лише два можливі типи "0" та "1". Мультиноміальна логістична регресія стосується ситуацій, коли результат може мати три або більше можливих типів (наприклад, "хвороба А" проти "хвороба В" проти "хвороба С"), які не впорядковані. Порядкова логістична регресія стосується впорядкованих залежних змінних. В системі була реалізована проста бінарна логістична регресія.

Використовуючи методи, які надає бібліотека scikit-learn користувачу надається можливість обрати залежну змінну, обрати вектор незалежних змін, обрати регуляризацію та її вплив. Обрати можна або регуляризацію Тихонова, або регресію Лассо. Віджет логістичної регресії реалізований в системі побудови сценаріїв представлено на рисунку 2.20.

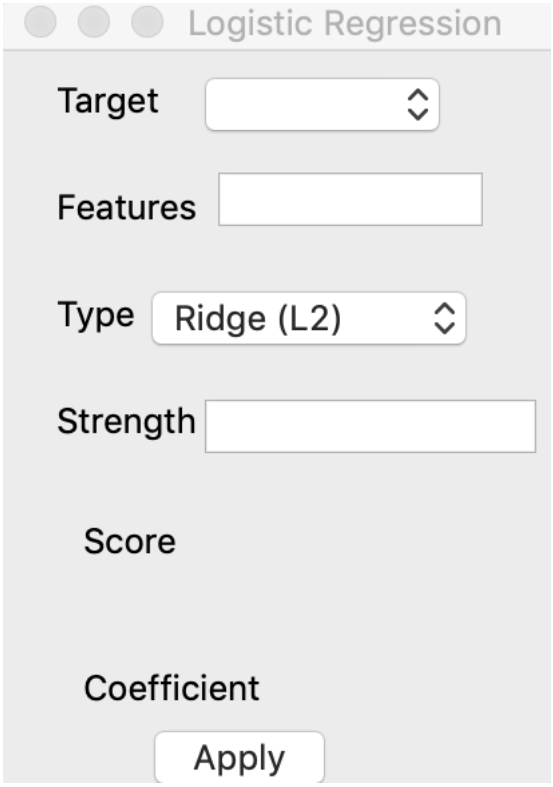


Рисунок 2.20 — Віджет для налаштування параметрів логістичної регресії

Наступний реалізований алгоритм це наївний баєсів класифікатор. Цей алгоритм являє собою ймовірнісний класифікатор, що використовує теорему Баєса для визначення із якою ймовірністю елемент вибірки приналежить до одного з класів при припущенні (наївному, від якого і йде назва алгоритму) незалежності змінних. У цього алгоритму є декілька доволі помітних переваг:

- хороші результати при роботі із наборами даних, де представлено більше двох класів даних;
- висока швидкість навчання;
- помірні вимоги до пам’яті, що може бути особливо критичне в роботі із великими даними.

Серед недоліків можна виділити наступні пункти:

- дата сет повинен мати усі приклади категорій, в інакшому випадку результати будуть не точні;
- ймовірність що дійсно усі параметри будуть незалежні майже неможливо досягнути в реальному житті[44].

В системі віджет роботи із налаштування цього класифікатора представлений доволі просто. Аналітику просто потрібно обрати залежні і незалежні параметри із вхідного набору даних.

Ще один алгоритм використаний в системі — метод опорних векторів. Основним завданням алгоритму є знайти найбільш правильну лінію, або гіперплоскость розділяє дані на два класи. SVM це алгоритм, який отримує на вході дані, і повертає таку розділяє лінію[45]. Алгоритм SVM влаштований таким чином, що він шукає точки на графіку, які розташовані безпосередньо до лінії поділу найближче. Ці точки називаються опорними векторами. Потім, алгоритм обчислює відстань між опорними векторами і розділяє площиною. Це відстань яке називається зазором. Основна мета алгоритму — максимізувати відстань зазору. Кращою гіперплоскостью вважається така гіперплоскость[46], для якої цей зазор є максимально великим. Однак на практиці доволі складно знайти набор даних, який було б легко розділити лінійно, тому у даного метода є багато варіацій із використанням різних функцій для задання плоскості. В системі було реалізовані наступні варіації алгоритму —лінійний, поліноміальний, сігмоїдальний та радіально-базисний SVM. Приклад віджету для роботи із лінійним SVM представлено на рисунку 2.21.

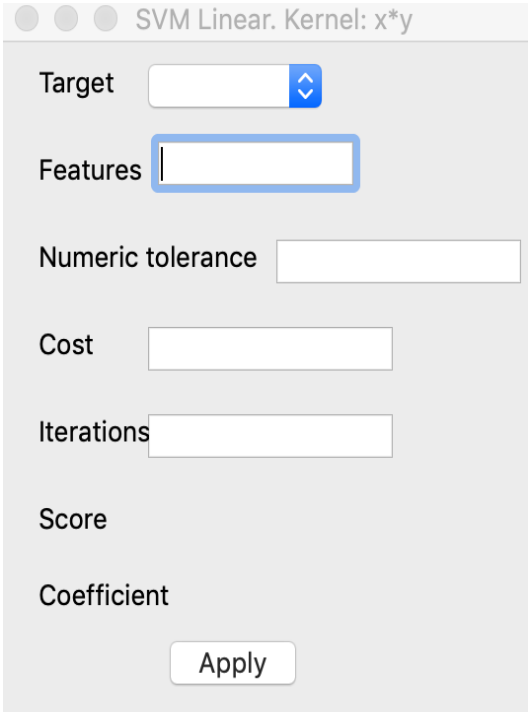


Рисунок 2.21 Віджет для налаштування лінійного SVM

Останнім алгоритмом, що був реалізований в системі — метод Random Forest. Це один з алгоритмів придуманий Лео Брейманом і Адель Катлер ще в минулому столітті[47]. Він використовуються для вирішення задач класифікації, регресії, кластеризації, пошуку аномалій, селекції ознак і т.д[48]. Тобто його можна застосувати до усіх задач, який є в інтелектуальному аналізі даних.

Метод Random forest — це безліч вирішальних дерев. У задачах регресії їх відповіді усереднюються, в завданнях по класифікації приймається рішення голосуванням за більшістю. Алгоритм можемо розглядати як серію питань так / ні про вхідних даних. В кінцевому підсумку питання призводять до передбачення певного класу (або величини в разі регресії). Питання про доступнні данні задається до тих пір, поки не буде знайдено певного рішення. Всі дерева будуються незалежно за наступною схемою:

- обирається підвибірка навчальної вибірки розміру samplesize, по якій будується дерево (для кожного дерева - своя підвибірка);
- для побудови кожного розщеплення в дереві переглядаємо max_features випадкових ознак (для кожного нового розщеплення - свої випадкові ознаки);
- алгоритм обирає найкращі ознака і розщеплення по ньому (по заздалегідь заданому критерію). Дерево будується, як правило, до вичерпання вибірки (поки в листі не залишаться представники тільки одного класу), але в сучасних реалізаціях є параметри, які обмежують висоту дерева, число об'єктів в листі і число об'єктів в підвибірки, при якому проводиться розщеплення[49].

Зрозуміло, що така схема побудови відповідає головному принципу ансамблюванню (побудови алгоритму машинного навчання на базі кількох, в даному випадку вирішальних дерев): базові алгоритми повинні бути хорошими і різноманітними (тому кожне дерево будується на своїй навчальній вибірці і при виборі розщеплення є елемент випадковості).

Цей алгоритм вимагає дуже багато ресурсів, а обмеження на глибину зашкодить точності (для вирішення складних завдань потрібно побудувати багато глибоких дерев). Можна помітити, що час навчання дерев зростає приблизно лінійно їх кількості.

В системі реалізований віджет для роботи із деревами що дозволяє налаштувати наступні критерії:

- залежні та незалежні зміні;
- максимальну кількість дерев;
- критерій для оцінювання якості розділення;
- максимальну глибину дерева.

Реалізація віджету представлена на рисунку 2.22.

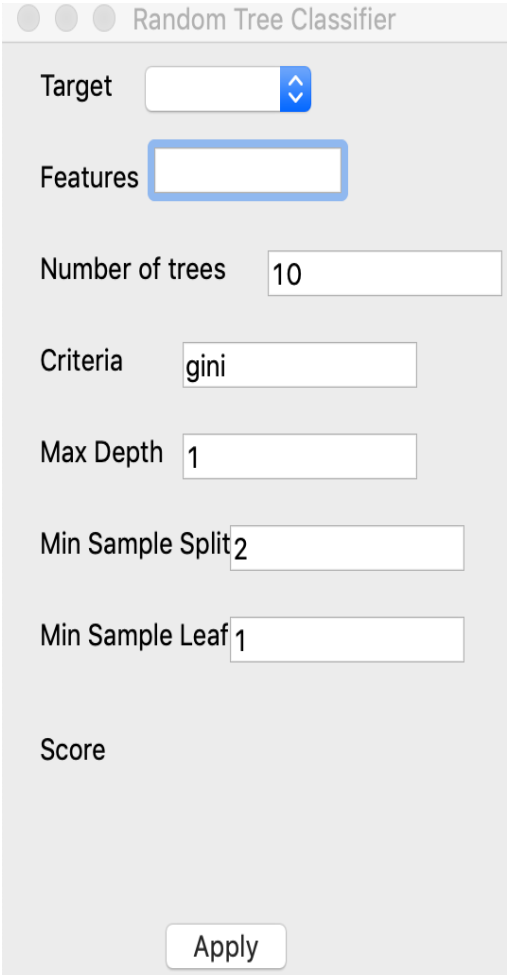


Рисунок 2.22 — Віджет для налаштування випадкового дерева

Останній віджет, який був реалізований в системі призначений для вирахування заданого параметру на інших дата сетах в залежності від отриманої моделі. Це стало можливо за допомогою функції predict, яка реалізована в усіх класах бібліотеки scikit-learn. Це дає практично оцінити ефективність розрахованих в алгоритмах коефіцієнтів на наборах даних, які були не задіяні при навчанні. Отриманий результат можна зберігати у csv файл, або візуалізувати.

2.4 Висновки до розділу 2

В даному розділі були розглянуті основні мови програмування, які застосовуються для роботи із великими даними та проведений порівняльний аналіз. Серед усіх обраних технологій лише Python задовольняє усім поставленим вимогам:

- багатий інструментарій для роботи із даними;
- багатий інструментарій для візуалізації;
- легкість в написані коду;
- легкість в підтримці та подальшому масштабуванні системи;

Серед усіх численних модулів, які надає мова програмування Python для реалізації були обрані бібліотеки pandas, numPy та scikit-learn для обробки та аналізу даних, та фрейморк PyQt для побудови графічного інтерфейсу із можливостями перетаскування віджетів (функціональних одиниць сценарію, що виконують лише одну конкретну дію) до робочого поля.

В ході роботи було розглянуті основні категорії дій, які необхідно мати в інструментальних засобах побудови сценаріїв аналітики великих даних та був реалізований фунціонал під кожну категорію. Для вхідних даних у систему були сформульовані чіткі критерії та обмеження, які необхідно притримуватися. Було продемонстровано реалізацію кожного із наявних в системі віджетів та параметри їх налащтування.

3. ПРИКЛАДИ РОБОТИ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ПОБУДОВИ СЦЕНАРІЇВ АНАЛІТИКИ ВЕЛИКИХ ДАНИХ

Для презентації роботи розроблених інструментальних засобів побудови сценаріїв аналітики великих даних буде продемонстрований на двох задачах. Буде описаний кожен крок для побудови типового сценарію, налаштування різних параметрів методів та алгоритмів і порівняння результатів отриманих моделей в залежності від вхідних параметрів.

3.1 Задача на регресію

Перший приклад буде посвячений передбаченню цін на нерухомість, оскільки таке прогнозування стає все більш і більш важливою. Ціни на житло є хорошим показником як загального стану ринку, так і економічного здоров'я країни. Подібна задача є однією із типових, з якими дозволяє впоратися інтелектуальний аналіз даних використовуючи засоби регресії. Для отримання результатів буде використаний алгоритми лінійної регресії та метод Random Forest Regression.

Обраний набір даних є прикладом із цін на нерухомість із різних міст Сполучених Штатів Америки. Збережена інформація в файлі формату csv. Кількість рядків у файлі — 21613. Складається дата сет із наступних сімнадцяти колонок:

- date —дату купівлі нерухомості, в форматі date;
- price — ціна, в форматі float;
- bedrooms — кількість спалень, в форматі int;
- bathrooms — кількість ванних кімнат, в форматі float. Значення 0.5 позначає кімнату де є вбиральню, проте немає душу;
- sqft_living — житлова площа, представлена в форматі float;
- sqft_lot — площа ділянки, представлена в форматі float;
- floors — кількість поверхів в будівлі, представлена в форматі float;

- waterfont — одна із колонок є фіктивною зміною у форматі int;
- view — оцінка стану будівлі від 0 до 4 в форматі int;
- condition — оцінка зовнішнього виду навколишньої території біля будівлі від 0 до 4 в форматі int;
- sqft_above — площа над земної частини будинку в форматі int;
- sqft_basement — площа підземної частини будинку в форматі int;
- yr_built — рік побудови в форматі int;
- yr_renovation — рік реновації в форматі int;
- lat— широта нерухомості в форматі float;
- long — довгота нерухомість в форматі float;
- statezip — поштовий індекс нерухомості представлений в форматі float ;

Перший крок, що потрібно виконати аналітику — завантажити файл. Оскільки це csv файл, а роздільником слугує кома, тому доцільно використовувати для цієї дії віджет “CSV”. Аналітик вікдриває вкладку “Read Data Activities”, обирає необхідний рядок клікає на нього і необхідна функціональна одиниця сценарію з’являється на робочому полі. Після цього необхідно поєднати віджет “CSV” та “Start”. Натиснувши ліву кнопку миші аналітик відкриє вікно налаштування віджету та вказує шлях до файлу та необхідний сепаратор. Результат цих маніпуляцій представлено на рисунку 3.1.

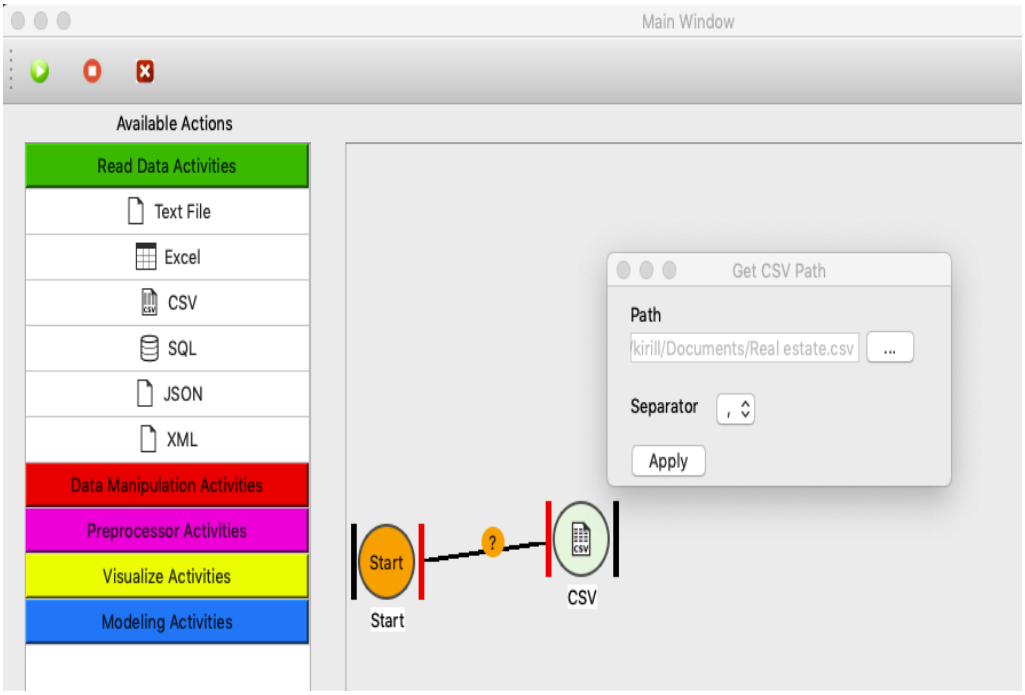


Рисунок 3.1 — Вигляд робочого вінка після вибору віджета завантаження

Оскільки в наборі даних, як було зазначено вище є одна фіктивна змінна waterfront, то є сенс видалити її з дата сету, щоб економити місце в пам'яті. Більш того немає ніякого сенсу в колонці, що відповідає за назву країни, оскільки всі данні відносяться до Сполучених Штатів Америки. Тому для цього можна використати віджет "Select Columns" із вкладки "Data Manipulation Activities". Вигляд налаштування віджету представлено на рисунку 3.2.

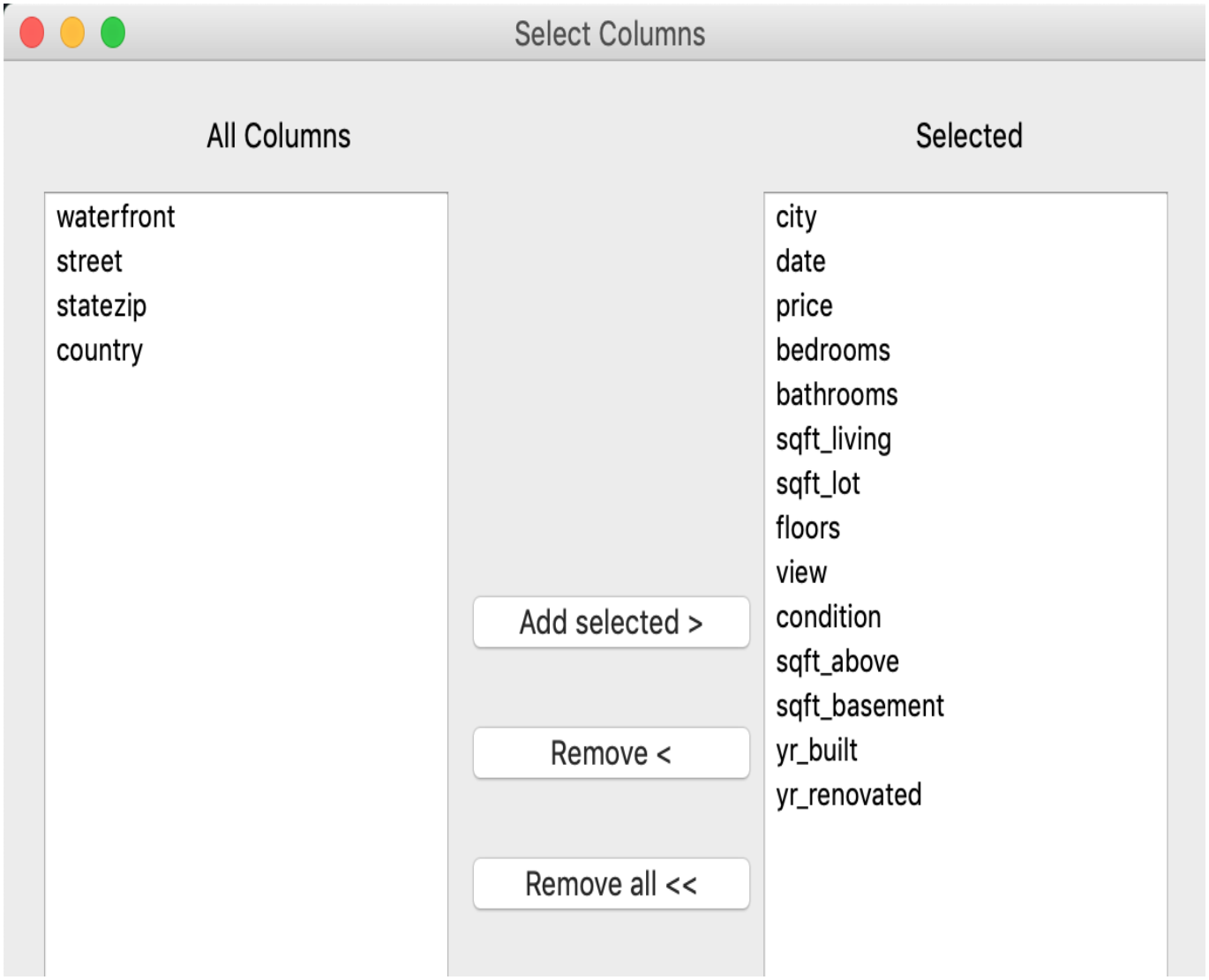


Рисунок 3.2 — Вибір стовпчиків із усього файл для подальшої побудови сценарію аналітики

Оскільки, як було зазначено вище, завжди є шанс що завантажені дані можуть мати "не чисті дані", то необхідно провести попередню обробку даних. Оскільки для цієї задачі був обраний дата сет без відсутніх даних має сенс перевірити чи наявні тут дублікати. Для цього із вкладки "Preprocessor activities" аналітик обирає віджет

“Remove Duplicates”. Ніяких додаткових параметрів для налаштування цей оператор не має, він просто видалить дані які уже були присутні в файлі.

Далі, аналітику необхідно обрати необхідні алгоритми для інтелектуального аналізу великих даних із вкладки “Modeling Activities” та поєднати із віджетом, що видаляв дублікати. Після цього додані оператори потрібно з’єднати із віджетом “Finish”, що означає закінчення сценарію. Вигляд побудованого сценарію представлено на рисунку 3.3.

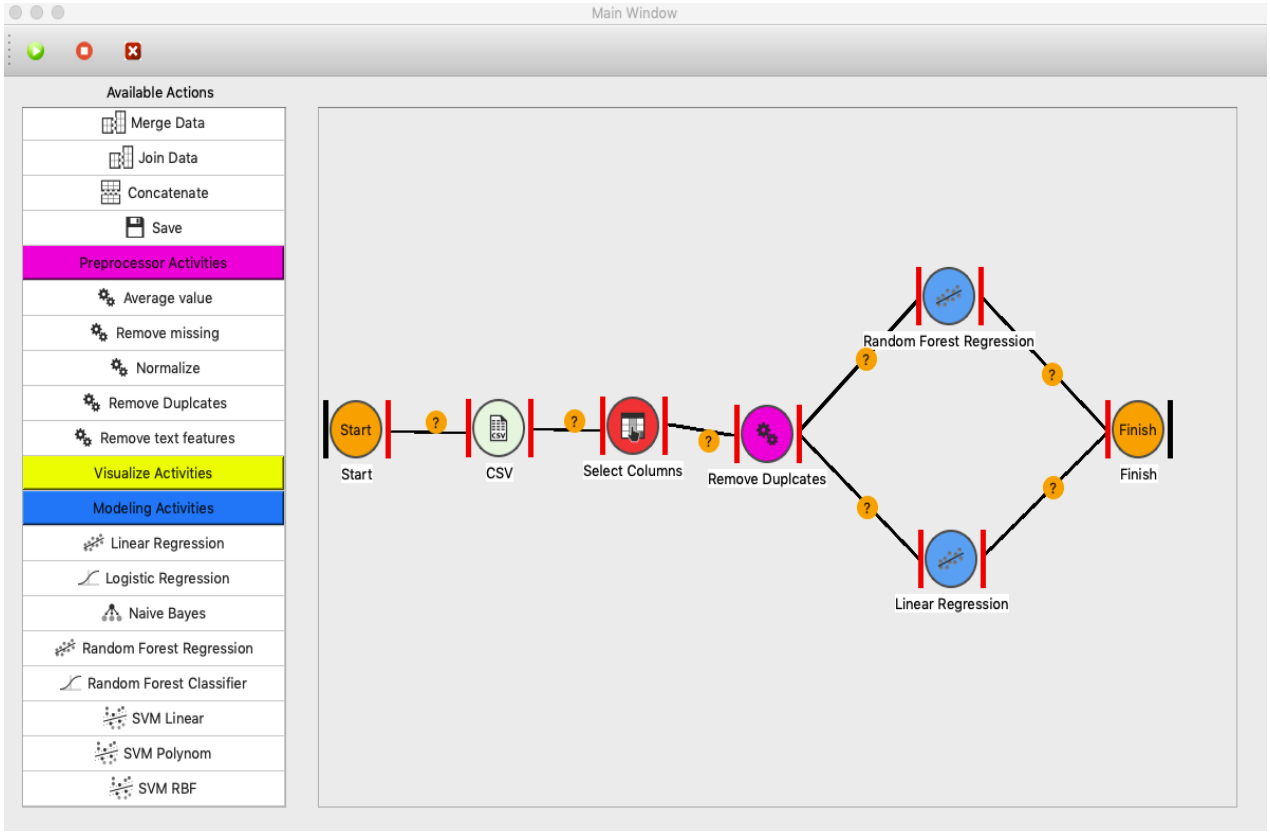


Рисунок 3.3 — Вигляд типового сценарію для задачі прогнозування цін на нерухомість

При пошуку оптимальної моделі аналітику необхідно обирати різні незалежні змінні та перевіряти їх результати, то необхідно подивитися різні результати та порівняти. Віджет для лінійної регресії додаткових налаштувань не потребують, а ось налаштування віджету для дерев буде наступною:

- кількість дерев буде дорівнювати 10;
- за критерій буде обраний показник MSE;
- максимальна глибина дерева буде обрана за 10;
- мінімальний показник для розбиття на ноди 5;

- мінімальний кількість для того, щоб бути нодою 2;

Оцінку ефективності алгоритмів забезпечує за допомогою вбудованого методу score із бібліотеки для роботи із великими даними sklearn. Він вираховує коефіцієт детермінації R^2. Результат 1 означає, що модель зазнала ефекту перенавчання, а 0 означає навпаки, що модель не дає адекватну модель. Результати проведеного експерименту забезпечуються завдяки методу представлено в таблиці 3.1.

Таблиця 3.1 Коефіцієнт детермінації при різних вхідних векторах незалежних змін

Вектор незалежних змін	Результати роботи лінійної регресії	Результати роботи алгоритму Random Forest
bedrooms,bathrooms,sqft_living	0.5	0.64
bedrooms,bathrooms,sqft_living,sqft_lot	0.5	0.69
bedrooms,bathrooms,floors,condition,sqft_lot,sqft_living	0,56	0,71
bedrooms,bathrooms,floors,condition,sqft_lot,sqft_living, grade,yr_built,view	0,63	0,86

3.2 Задача на класифікацію

Наступний приклад буде присвячений задачі класифікації. Подібні задачі є одними із найпоширеніших в інтелектуальному аналізі даних. Завдання класифікації має на меті передбачити до якої категорії буде належати даний немаркований елемент з набору даних. Категорія, або клас, повинен бути обраний серед обмеженого набору попередньо визначених класів. Найпростіший тип подібних задач — це бінарна класифікація. У ній цільовий атрибут має лише два можливі значення: наприклад, шукана подія могла статися або навпаки не могла. Багатоцільові цілі мають більше двох значень: наприклад подія могла статися із вірогідністю в нуль, двадцять, сорок, шістдесят, вісімдесят або сто відсотків.

Обрана для прикладу задача відноситься до простої бінарної класифікації і використовуватися для цього буде набір даних пов’язаний із американською баскетбольною лігою NBA. Суть задачі полягає в тому, щоб базуючись на статистиці виступів новачків побудувати модель, що буде класифікувати, чи будуть вони грати більше п’яти років чи ні та перевірити результати отриманої моделі на невеликому наборі даних.

Дата сети були звантажені у форматі csv та використовують кому, як роздільник. Кількість рядків у файлі, що буде використовуватися для побудови моделі — 1200. У файлі, на якому будуть перевіряти результати моделі — 200. Кожен набор даних складається із наступних 21 колонок:

- Name — ім’я гравця в форматі строки;
- GP — кількість зіграних ігор;
- MIN — кількість хвилин зіграних за матч в середньому;
- PTS — калькість очок отриманих за матч в середньому;
- FGM — кількість забитих м’ячів в середньому;
- FGA — кількість спроб забити м’яч в середньому;
- FG% — відношення забитих м’ячів до загальної кількості спроб;
- 3P Made — кількість забитих трьох очкових в середньому;
- 3PA — кількість спроб забити трьох очкових в середньому;
- 3P% — відношення забитих трьох очкових до загальної кількості спроб;
- FTM — кількість забитих штрафних кидків;
- FTA — кількість спроб забитих штрафні кидки;
- FT% — відношення забитих штрафних кидків до загальної кількості спроб;
- OREB — кількість підборів в нападі в середньому;
- DREB — кількість підборів в захист в середньому;
- REB — загальна кількість підборів в середньому;
- AST — кількість асистів в середньому;
- STL — кількість відборів м’яча за гру с середньому;

- BLK — кількість блоків за гру в середньому;
- TOV — кількість втрат за гру в середньому;
- TARGET_5Yrs — значення чи була кар'єра гравця довша за 5 років. Цифра 0 означає ні, цифра 1 — так.

Отже перше, що буде необхідно зробити аналітику завантажити два файли. Для цього він може використати віджет "CSV" "Read Data Activities". Після цього необхідно провести попередню обробку даних, оскільки на сайті звідки було завантажено дані зазначено, що не усі строчки маються значення, то необхідно видалити рядки де є пропуски. Для цього із вкладки "Preprocessor Activities" потрібно обрати віджет "Remove missing" для кожного csv файлу окремо. Після цього аналітик може обрати методи для обробки даних. Нехай в цій задачі це буде алгоритм опорних векторі, що використовує радіальну базисну функцію та метод Random Forest Classifier.

Налаштування віджету "SVM RBF" будуть наступні:

- параметр толерантності дорівнюю 0.01 ;
- параметр толерантності класифікації буде дорівнювати 2;
- кількість ітерацій 10000;

Налаштування віджету "Random Forest Classifier" наступні:

- кількітсть дерев 15;
- буде використовуватися коефіцієнт Джині;
- глибина дерева 15;
- мінімальний показник для розбиття на ноди 7;
- мінімальний кількість для того, щоб бути нодою 3;

Після цього результати моделей можна перевірити за допомогою віджету "Predict". Вхідними значеннями для цього віджету буде модель та дата сет для перевірки. Після цього використовуючи функцію predict до набору даних для перевірки додається нова колонка, де буде зберігатися вирахуване значення на основі вхідної моделі. Зберегти результати можно у вигляді CSV файлу використовуючи віджет "Save". Вигляд сценарію представлений на рисунку 3.4.

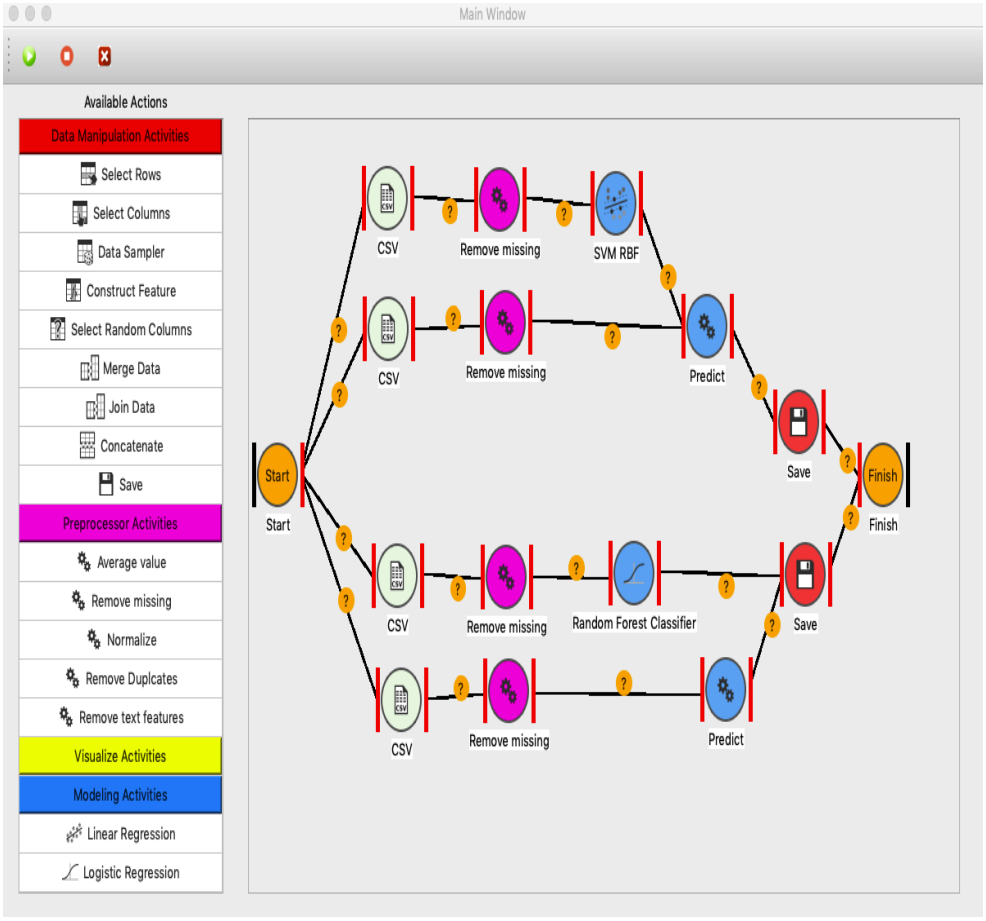


Рисунок 3.4 — Сценарій порівняння двох методів при задачі бінарної класифікації

Результати проведеного експерименту забезпечуються завдяки методу представлено в таблиці 3.2.

Таблиця 3.2 Коефіцієнт детермінації при різних вхідних векторах незалежних змін

Вектор незалежних змін	Результати роботи SVM RFB	Результати роботи алгоритму Random Forest
BLK, GP, REB, 3P Made	0.63	0.69
MIN, 3P%, FG%, FT%	0.65	0,73
MIN, 3P Made, PTS	0.71	0.79
BLK, GP, MON, PTS	0,83	0,88

3.4 Висновки до розділу 3

В цьому розділі були представлені результати, які можна отримати використовуючи інструментальні засоби для побудови сценаріїв аналітики великих даних. Для цього було обрано дві задач, одна з яких була присвячена регресії, а інша класифікації.

Одна з типових задач в яких використовується регресія — це передплачування якогось значення на основі вектору незалежних змінних. В якості прикладу був обраний набір значень із цінами на житло у Сполучених Штатах Америки, що складався із приблизно 21000 строк та 17 колонок. Провівши необхідні кроки для попередньої обробки даних, було обрано два алгоритми для побудови моделі. За результатам коефіцієнту детермінації можна побачити, що алгоритм Random Forest давав набагато кращі прогнози при однаковому вхідному векторі незалежних змін.

Друга задача була присвячена розгляду простої бінарної класифікації. Мета задачі була визначити чи буде кар’єра гравця американської баскетбольної ліги NBA довша за 5 років, та перевірити модель на невеликому файлі із 200 рядків. Для цього було обрано два алгоритми — алгоритм Random Forest та метод опорних векторів, що використовує радіально базисну функцію. Не зважаючи на те, що більшість результатів вказувало на перевагу алгоритму Random Forest, при певному наборі вхідного вектору різниця була не велика. А оскільки побудова випадкового лісу доволі затратний набір операцій для процесору, то це дозволяє аналітику обрати альтернативний засіб, що може зекономити час і отримати результати значно швидше.

4. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ “ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ПОБУДОВИ СЦЕНАРІЇВ АНАЛІТИКИ ВЕЛИКИХ ДАНИХ”

В даному розділі дисертації буде описано економічне обґрунтування реалізації стратап-проекту на тему “Інструментальні засоби побудови сценаріїв аналітики великих даних”. Метою цього розділу є розвиток та формування інноваційного мислення, підприємницького духу, що відповідає стандартам часу, та формування здібностей до оцінювання перспектив і можливостей комерціалізації науково-технічних розробок на сучасному ринку програмного забезпечення в умовах висококонкурентної ринкової економіки. Основою для стартап-проекту стали сформовані у попередніх розділах дисертації концепції “Інструментальних засобів побудови сценаріїв аналітики великих даних”.

4.1 Опис ідеї стартап проекту

В межах даного підпункту буде проаналізовано та подані у вигляді таблиць наступні концепції:

- зміст ідеї, що розлядається;
- потенційні напрямки застосування;
- основні вигоди, які отримає користувач програмного забезпечення;
- основні відмінності від існуючих аналогів на ринку;

Перші три пункти списку представлено у таблиці 4.1 та надаються цілісне уявлення про зміст ідеї і можливі потенційні ринки, де може бути проданий стартап.

Таблиця 4.1 Опис ідеї стартап-проекту

Зміст ідеї	**Напрямки застосування**	**Вигоди для користувача**

Таблиця 4.1 Опис ідеї стартап-проекту (продовження)

Побудова сценаріїв аналітики великих даних	Використовувати в інформаційно-аналітичні системи	Використовувати, як модуль для побудови сценаріїв аналітики великих даних при побудові інформаційно-аналітичної системи
	Використовувати для навчання у ВНЗ	Демонстрація роботи алгоритмів інтелектуального аналізу даних у вищих навчальних закладах
	Використовувати для побудови сценарії аналітики великих даних	Будувати сценарії аналітики великих даних на базі завантажених даних

Отже, проект може бути використаним як для аналітиків при роботі із різними проблемними областями в різноманітних бізнес сферах, так і для демонстрації студентам у вищих навчальних закладах принципи роботи алгоритмів інтелектуального аналізу даних.

Аналіз потенційних техніко-економічних переваг ідеї, тобто перелік пунктів чим відрізняється товар від існуючих аналогів, порівняно із пропозиціями конкурентів включає в себе наступне:

- визначення списку техніко-економічних властивостей та характеристик ідеї [26];
- визначення попереднього кола проектів розроблених конкурентами або продуктів-замінників і товарів-аналогів, що вже присутні на ринку, та проводиться

пошук інформації щодо значень техніко-економічних показників для ідеї власного стартап-проекту та проектів-конкурентів відповідно до визначеного вище переліку;

- проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші показники та позначаються як W (що означає слабкі); б) аналогічні тобто нейтральні значення, позначаються як N; в) та кращі значення, які позначаються S. Результати можна побачити в таблиці 4.2.

Таблиця 4.2 Визначення слабких, нейтральних та сильних характеристик ідеї проекту

№ п/п	**Техніко-економічні характеристики ідеї**	**(Потенційні) товари/концепції конкурентів**				**W**	**N**	**S**
		Мій проект	**Конкурент 1**	**Конкурент 2**	**Конкурент 3**			
1	Форма виконання	Дода-ток	Додаток	Додаток	Додаток		+	
2	Собівартість	Низька	Висока	Висока	Середня			+
3	Масштабова-ність	Так	Ні	Ні	Ні			+
4	Наявність інтернету	Ні	Так	Так	Так			+
5	Кроспалтфор-меність	Так	Так	Так	Ні		+	
6	Наявність адміністратора для налаштування	Так	Ні	Ні	Так	+		

Проект має сильні сторони, що відсутні в існуючих аналогів тому здатний з ними конкурувати. Серед сильних сторін, це низька собівартість, проста масштабованість та відсутність потреби підключатися до мережі Інтернет. Масштабованість є основною перевагою і дозволяє розширяти проект новим функціоналом без надмірних зусиль підлаштовуючи програмне забезпечення під потреби клієнтів індивідуально. Слабкою стороною є необхідність мати адміністратора, який би забезпечив машину, на якій буде встановлені необхідні пакети та бібліотеки мови програмування Python, як sklearn, pandas, numpy та pyqt.

4.2 Технологічний аудит ідеї проекту

В межах даного підрозділу пройде аудит технології, за допомогою якої можна реалізувати ідею. Визначення технологічної передбачає аналіз таких складових:

- за якою технологією буде виготовлено товар згідно ідеї проекту;
- чи існують такі технології, чи їх потрібно розробити/доробити;
- чи доступні такі технології авторам проекту.

Результати наведено в таблиці 4.3.

Таблиця 4.3 Технологічна можливість реалізації проекту

No п/п	**Ідея проекту**	**Технології її реалізації**	**Наявність технологій**	**Доступність технологій**
1	Прикладний програмний інтерфейс	Фреймоврк PyQt мови Python	Наявна	Доступна безкоштовно
2	Алгоритми для роботи із великими даними	Бібліотеки scikit-learn, pandas, numpy мови програмування Python	Наявна	Доступна безкоштовно
Висновок: проект реалізувати можливо (бо я його зробив)				

Таблиця 4.3 Технологічна можливість реалізації проекту (продовження)

Обрана технологія реалізацій: настільний додаток з інтерфейсом, що підтримую функціонал перетягування, алгоритми для інтелектуального аналізу даних.

У Таблиці 4.3 надано результати огляду основних стеків технологій, що можуть бути використані з метою реалізації системи стартап-проекту описаного в минулих розділах дисертації. Було обрані технології що не потребують доопрацювання.

4.3 Аналіз ринкових можливостей

Визначення ринкових можливостей, які можна використати під час ринкового впровадження розробленого програмного забезпечення та потенційних загроз, які можуть перешкодити реалізації стартап-проекту, дозволяє спланувати напрями розвитку компанії з урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій конкурентів. Перший крок, що буде проведений в магістерський дисертації — це аналіз попиту, його наявність, обсяг та динаміка розвитку ринку. Результати наведені в таблиці 4.4.

Таблиця 4.4 Попередня характеристика потенційного ринку стартап-проекту

No п/п	**Показники стану ринку (найменування)**	**Характеристика**
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	20000 грн

Таблиця 4.4 Попередня характеристика потенційного ринку стартап-проекту(продовження)

3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікацій	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	R = (3000000*100) /(1000000*12) = 25%

За результатами аналізу таблиці робиться висновок щодо того, чи є ринок привабливим для входження за попереднім оцінюванням. Обмежень для входу на ринок відсутні, динаміка ринку постійно зростає, галузь є дуже рентабельною.

Наступний крок це визначення потенційної групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи конкретної категорії. Результати представлені в таблиці 4.5.

Таблиця 4.5 Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінність у поведінці різних потенційних цільових груп	Вимоги споживачів до товару
1	Сервіс для побудови сценаріїв аналітики великих даних	Будь-які підприємства що надають послуги; ВНЗ, де є предмет пов'язанні з аналітикою великих даних	Відмінність полягає у тому, що для підприємства можливо буде потрібно реалізувати функціонал підходящий під специфіку його діяльності. Для ВНЗ ж буде необхідно лише реалізація алгоритмів відповідно до навчальної програми, які зазвичай покриваються усі стандартні алгоритми.	Стабільність роботи Невисока ціна Наявність пробного періоду Наявність документації Точність роботи

Після визначення потенційних груп клієнтів необхідно провести аналіз ринкового середовища: скласти таблиці різних факторів, що сприяють ринковому впровадженню програмного забезпечення, та факторів, що можуть стати на заваді вдалої реалізації. Результати наведені в таблиці 4.6 та таблиці 4.7.

Таблиця 4.6.Фактори загроз

No п/п	**Фактор**	**Зміст загрози**	**Можлива реакція компанії**
1	Конкуренція	Вихід на цільовий ринок програмного забезпечення великої компанії	Вийти з цільового ринку та шукати інший; Анонсувати новий функціонал власного програмного забезпечення в момент виходу конкурента; Зробити акції на ціни в момент виходу конкурента;
2	Вимоги користувачів можуть змінитися	Інструмент обмежений наявними функціями, не відповідає актуальним вимогам ринку і не має деяких функцій, які мають конкуренти	Реалізація та анонс нового функціоналу в найкоротші строки.

Таблиця 4.7 Фактор можливостей

No п/п	**Фактор**	**Зміст можливості**	**Можлива реакція компанії**
1	Зростання фінансових можливостей у потенційних клієнтах	Зростання фінансування ВНЗ або підприємтс, які займаються наданням послуг	Запропонувати їм свій продукт за бонусною ціною для зацікавлення
2	Поява нових методів інтелектуального аналізу даних	Поява в бібліотеках для data maning, що використовуються в програмному забезпечені нових методів та алгоритмів, що надають кращий результат.	Реазлізувати ці методи в програмному забезпечені, щоб мати перевагу перед конкурентами.

Наступний крок, що необхідно зробити — провести аналіз пропозиції, в ході якого будуть визначені загальні риси конкуренції на обраному ринку. Результати представлені в таблиці 4.8.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	**В чому проявляється дана характеристика**	**Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)**
1. Вказати тип конкуренції — досконала	На ринку присутні 3 фірми-конкуренти	Цінова політика та послуги, які надає програмне забезпечення конкурентів, повинні бути враховані на початкових етапах розробки.
2. За рівнем конкуренції — міжнародна	Усі компанії конкуренти не з України.	Передбачити можливість обирати мову проекту, щоб на при виході на міжнародний ринок не було проблем.
3. За галузевою ознакою -— міжгалузева	Конкуренти мають ПЗ, яке використовується у різних галузях.	Проаналізувати програмне забезпечення конкурентів в усіх сферах, виділити недоліки та розробити свій проект без них.
4. Конкуренція за видами товарів — товарно-видова	Товар для усіх фірм є однаковий, саме програмне забезпечення	Створити свій товар враховуючи мінуси конкурентів.
5. За характером конкурентних переваг — нецінова	Зробити собівартість проекту нижчою ніж у конкурентів	Використання менш дорогих технологій для розробки, аніж ті що обрали конкуренти.
6. За інтенсивністю - не марочна	марочна	-

У таблиці 4.8 був наведений ступеневий аналіз конкуренції на ринку, в якому було визначено конкурентне середовище та його особливості, а також подальший їх вплив на діяльність компаніях при розробці стартапу. Після цього необхідно провести більш детальний аналіз умов конкуренції. Результати представлений в таблиці 4.9.

Таблиця 4.9 Аналіз конкуренції в галузі за М. Портером

Складові аналізу	**Прямі конкуренти в галузі**	**Потенційні конкуренти**	**Постачальники**	**Клієнти**	**Товари-замінники**
	Навести перелік прямих конкурентів	**Визначити бар'єр входження в ринок**	**Визначити фактори сили постачальників**	**Визначити фактори сили споживачів**	**Фактори загроз з боку замінників**
Висновки	На ринку аналітики наявні 3 конкуренти. Найбільш схожим на стартап-проект є конкурент номер 1.	Можливість виходу на ринок присутня, бо рішення є масштабованним і легко підлаштовується під потреби нових клієнтів	Постачальники відсутні	Важливим для користувача є зручність та наявність необхідного функціоналу під конкретну сферу	Товари-замінники можуть використати дешевшу технологію при створенні ПЗ та зменшити собівартість кінцевого результату.

На основі аналізу конкуренції та урахуванням характеристик ідеї стартап-проекту, факторів ринкового середовища та вимог і потреб потенційних клієнтів визначається та обґрунтовується перелік факторів конкурентоспроможності.

За результатами аналізу необхідно зробити висновок щодо принципової можливості роботи на ринку програмного забезпечення враховуючи до уваги на сформовану конкуренту ситуацію. Також потрібно зробити висновок щодо характеристик (тобто сильних сторін стартап-проекту), які необхідно реалізувати в програмному забезпеченні, для того щоб бути конкуренто спроможнім на обраному

ринку. Враховуючи вище проведені аналізи та їх результати, формується таблиця 4.10, що визначає фактори конкурентоспроможності.

Таблиця 4.10 Обґрунтування факторів конкурентоспроможності

No п/п	**Фактор конкурентоспроможності**	**Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)**
1	Масштабованість системи	Дозволяє підлаштовувати систему під потреби кожного конкретного замовника.
2	Безпечність	Додаток не буде збирати особисту інформацію користувача

За визначеними факторами конкурентоспроможності потрібно далі провести аналіз сильних та слабких сторін стартап-проекту сформованих в таблиці 4.11

Таблиця 4.11 Порівняльний аналіз сильних та слабких сторін

No п/п	**Фактор конкурентоспроможності**	**Бали 1-20**	**Рейтинг товарів-конкурентів у порівнянні з даним продуктом**						
			-3	**-2**	**-1**	**0**	**1**	**2**	**3**
1	Масштабованість системи	20	+						
2	Безпечність	10			+				

Останнім кроком ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних та слабких сторін, загроз та можливостей). Результати представлені в таблиці 4.12.

Таблиця 4.12 SWOT аналіз стартап-проекту

Сильні сторони: Масштабованість	**Слабкі сторони:** Необхідність мати адміністратора для налаштування
Можливості: Додаткове фінансування у потенційних покупців	**Загрози:** Конкуренція та можливі зміни потреб

На основі SWOT-аналізу потрібно розробити альтернативи ринкової поведінки (що буде являти собою перелік заходів) для виведення стартап-проекту на ринок та

орієнтовний оптимальний час реалізації з огляду на програмне забезпечення яке надають конкуренти. Визначені альтернативи потрібно проаналізувати за двома критеріями: строки та ймовірності отримання ресурсів. Результати наведені в таблиці 4.13.

Таблиця 4.13 Альтернативи впровадження стартап-проекту

No п/п	**Альтернатива (орієнтовний комплекс заходів) ринкової поведінки**	**Ймовірність отримання ресурсів**	**Строки реалізації**
1	Створення програмного забезпечення на основі бібліотеки scikit-learn для інтелектуального аналізу даних	85%	6 місяців
2	Створення програмного забезпечення без використання бібліотек для інтелектуального аналізу даних	25%	18 місяців

З означених альтернатив обирається та, для якої: а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – більш стислими. Тому обираємо альтернативу 1.

4.3 Розроблення ринкової стратегії продукту

Перший крок для побудови ринкової стратегії стає опис цільових груп потенційних споживачів. Результати аналізу представлені в таблиці 4.14.

Таблиця 4.14 Вибір цільових груп потенційних споживачів

No п/п	**Опис профілю цільової групи потенційних клієнтів**	**Готовність споживачів сприйняти продукт**	**Орієнтовний попит в межах цільової групи (сегменту)**	**Інтенсивність конкуренції в сегменті**	**Простота входу у сегмент**

1	ВНЗ	Масштабованість не є пріоритетним у даному випадку.	Дослідження в області великих даних проводяться постійно	Існують 3 конкурента, але вони не мають такої масштабованості	У сегмент увійти непросто, бо бюрократія всередині університет і є занадто складною.
2	Дослідницькі центри	Масштабованість системи додає їм можливості налаштувати систему під себе та виконати більшу кількість замовлень.			Завдяки можливості налаштувати систему під вимоги замовника — проста
3	Підприємства				
Як цільові групи були обрані дослідницькі центри та підприємства					

Далі для роботи в обраних сегментах ринку необхідно визначити основну стратегію розвитку програмного продукту. За М. Портером, існують три базові стратегії розвитку (лідерства, диференціації та спеціалізації), які відрізняються за такою характеристикою як ступінь охоплення цільового ринку та типом конкурентної переваги. Вона відрізняються за такими ключовими характеристиками, як витратами та визначними якості послуги чи товару, яку забезпечує стартап-проект. Результати обраної стратегії представлено в таблиці 4.15.

Таблиця 4.15 Визначення стратегії розвитку

No п/п	**Обрана альтернатива розвитку проекту**	**Стратегія охоплення ринку**	**Ключові конкурентоспро-можні позиції відповідно до обраної альтернативи**	**Базова стратегія розвитку***

Таблиця 4.15 Визначення стратегії розвитку(продовження)

1	Створення програми з використання бібліотеки scikit-learn.	Ринкове позиціювання	Масштабованість, безпечність	Стратегія спеціалізації (спирається на диференціа-цію)

Наступним кроком є вибір стратегії конкурентної поведінки серед чотирьох основних запропонованих М. Портером. Це стратегії лідеру, стратегія виклику лідеру, стратегія наслідування лідера та стратегія заняття конкурентної ніші. Результати представлені в таблиці 4.16.

Таблиця 4.16 Визначення базової стратегії конкурентної поведінки

No п/п	**Чи є проект «першопрохідним» на ринку?**	**Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?**	**Чи буде компанія копіювати основні характеристики товару конкурента, і які?**	**Стратегія конкурентної поведінки***
1	Ні	Так	Буде, а саме: основна задача створити масштабовану та безпечну систему для побудови сценаріїв аналітики великих даних	Стратегія заняття конкурентної ніші

З обраних вимог до постачальника, саме стартап-компанії та до кінцевого програмного забезпечення розробляється стратегія позиціонування. Її зміст полягає у формуванні ринкової позиції, за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.17 Визначення стратегії позиціювання

No п/п	**Вимоги до товару цільової аудиторії**	**Базова стратегія розвитку**	**Ключові конкуренто-спроможні позиції власного стартап-проекту**	**Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)**
1	Стабільність роботи Невисока ціна Наявність документації Підтримка багатьох платформ	Стратегія спеціалізації (спирається на диференціацію)	Можливість масштабувати систему під специфіки певної галузі	Надійність, безпека, простота в освоєнні

4.5 Розробка маркетингової програми проекту

Перше, що потрібно зробити — це підсумувати результати попереднього аналізу конкурентоспроможності товару. Результат аналізу представлений в таблиці 4.18.

Таблиця 4.18 Визначення ключових переваг концепції потенційного товару

No п/п	**Потреба**	**Вигода, яку пропонує товар**	**Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)**
1	Масштабова-ність	Можливість налаштувати систему під певну галузь	Більшість конкуренти не надають можливість адаптувати рішення під конкретну сферу застосування.

Таблиця 4.18 Визначення ключових переваг концепції потенційного товару(продовження)

2	Безпека	Не ведеться ніяке зберігання приватних даних в середині системи.	Користувачеві немає потреби замислюватися чи може бути якась загроза їх особистим даним, чи особисті даним клієнтів які вони використовують для побудови сценарію аналітики.

Надалі розробляється трирівнева маркетингова модель товару: уточняються ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання. Перелік кожного рівня наступний:

- 1-й рівень. При формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби або проблеми буде фінальний результат, яка його майбутня вигода. Дане питання безпосередньо пов’язане з формуванням технічного завдання в процесі розробки;
- 2-й рівень. Цей рівень являє рішення того, як буде реалізований товар включи в себе якість, властивості, дизайн;
- 3-й рівень. Товар з підкріпленням — додаткові послуги та переваги для споживача, що створюються на основі товару за задумом і товару в реальному виконанні. [49] [50]

Результат представлений в таблиці 4.19.

Таблиця 4.19 Опис трьох рівнів моделі товару

Рівні товару	**Сутність та складові**		
I. Товар за задумом	Інструментальні засоби для побудови сценаріїв аналітики великих даних		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Масштабованість системи	Нм	Технологічне

	Простота інтерфейсу	Нм	Технологічне
	Безпека	Нм	Технологічне
	Якість: згідно до стандарту ISO 4444 буде проведено тестування		
	Маркування відсутнє		
	Моя компанія. “KDMA”		
III. Товар із підкріпленням	1-місячна пробна безкоштовна версія. Постійна підтримка для користувача.		
За рахунок чого потенційний товар буде захищено від копіювання: патент.			

В ході написання роботи було наведено три рівні моделі товару, з яких можна зробити наступний висновок — основні властивості товару у реальному виконанні є нематеріальними та технологічними. Також була сформована сутність та складові товару у задумці та товару з підкріпленням

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари-аналоги або товари замінники, а також аналіз рівня доходів обраної на попередніх етапах групи споживачів. Результати можна побачити в таблиці 4.20.

Таблиця 4.20 Визначення меж встановлених цін

No п/п	**Рівень цін на товари-замін-ники, грн.**	**Рівень цін на товари-аналоги, грн.**	**Рівень доходів, грн**	**Верхня та нижня межі встановлення ціни на товар/послугу, грн.**
1.	50000	60000	400000	40000

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення, яке можна побачити в таблиці 4.21.

Таблиця 4.21 Формування системи збуту

No п/п	**Специфіка закупівельної поведінки цільових клієнтів**	**Функції збуту, які має виконувати постачальник товару**	**Глибина каналу збуту**	**Оптимальна система збуту**
1	Клієнт повинен надаватися в режимах “безкоштовної пробної версії” та “повна”. Продовження ліцензії робиться щорічно після оплати.	Продаж	0 (напряму), 1 (через одного посередника)	Власна та через посередників.

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів. Результати збережені в таблиці 4.22.

Таблиця 4.22 Концепції маркетингових комунікацій

No п/п	**Специфіка поведінки цільових клієнтів**	**Канали комунікацій, якими користуються цільові клієнти**	**Ключові позиції, обрані для позиціонування**	**Завдання рекламного повідомлення**	**Концепція рекламного звернення**
1	Купівля через веб-сайт компанії	Інтернет	Підтримка декількох платформ Пришвидшення розробки ПЗ	Показати переваги сервісу, у тому числі перед конкурентами	Демонстра-ційне ві-део із викорис-танням системи

4.6 Висновки до розділу 4

Згідно до проведених досліджень існує можливість ринкової комерціалізації проекту. Також, варто відзначити, що існують перспективи впровадження з огляду на потенційні групи клієнтів, бар‘єри входження не є дуже високими у певних сферах, а проект має дві значних переваги перед програмним забезпеченням конкурентів. Для успішного виконання проекту необхідно реалізувати програму із використанням засобів мови програмування Python.

Для успішного виконання проекту необхідно реалізувати інструментальні засоби побудови сценаріїв аналітики великих даних. В рамках даного розділу були обчислені основні фінансово-економічні показники стартап-проекту, а також проведений менеджмент можливих ризиків. Проаналізувавши отримані результати, можна вважати, що подальша реалізація є доцільною.

Наявні такі фактори загроз як — конкуренція, зміна потреб користувачів, зміна тарифів в існуючих гравців на ринку, виходу на ринок альтернативного програмного забезпечення з меншою собівартістю, потенційне уповільнення росту ринку.

ВИСНОВКИ

Основним завданням даної дисертації було створення інструментальних засобів побудови сценаріїв аналітики великих даних для аналітиків для роботи у різних проблемних областях. Актуальність такої задачі є досить вагомою через велику кількість наукових досліджень, так і через великий потенціал використання у комерційних проектах.

Було проведено огляд сценарного підходу в аналітиці великих даних, розкрито зміст та виділені основні кроки, які виконуються аналітиком або групою аналітиків при побудові сценарію у різних проблемних областях. Було розкрито, як сучасні методи та алгоритми дозволяють будувати сценарії для пошуку знань та кореляцій, які не можливо було отримати раніше.

В ході виконання роботи був також проведений аналіз вже існуючого програмного забезпечення для побудови сценаріїв, проаналізовано його функціонал, переваги та недоліки. Це дозволило окреслити головні характеристики майбутнього додатку, який був розроблений в подальшому виконанні дисертації.

Для побудови програмного забезпечення було проведено порівняльний аналіз існуючих технологій та мов програмування, які широко використовуються у сфері роботи із великими даними. На його основі було вирішено використовувати мову програмування Python, як для розробки графічного інтерфейсу застосовуючи фрейм ворк PyQT, так і для процесу інтелектуального аналізу даних використовуючи бібліотеки scikit-learn, numpy та pandas.

Відповідно до прийнятих рішень в процесі вивчення функціоналу вже існуючих інструментальних засобів для побудови сценаріїв аналітики великих даних, у практичній роботі були виведені наступні п’ять основних категорій функціональних операторів:

- оператори для зчитування даних;

- оператори для маніпуляціями з даними;
- оператори для попередньої обробки даних;
- оператори для аналізу даних;
- оператори для візуалізації даних;

В кожен підпункт була реалізована певна кількість операторів, що покривають основний функціонал, який необхідно мати в системі для побудови сценаріїв. Кожен такий оператор в системі називається віджет. В ході виконання роботи були сформовані вимоги до файлів різного формату, що будуть джерелами даних для системи. Також були продемонстровані приклад налаштування віджетів кожного типу та описана специфіка їх роботи.

Для демонстрації результатів, що система дійсно може використовуватися для побудови сценаріїв аналітики було розглянуто дві задачі. Одна була присвячена передбаченню цін на житло, тобто типова задача регресії, а інша — простій бінарній класифікації. Було продемонстровані результати в залежності від різних параметрів самих алгоритмів, так і векторів незалежних змінних, що обиралися.

В рамках розробки стартап-проекту було визначено перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару, що є підґрунтям для формування його конкурентоспроможності; обрана технологія реалізації ідеї проекту: для створення клієнт-серверного додатку були обрані технології мови Python, яка є безкоштовною та з якою мають досвід роботи члени проекту; проведений ступеневий аналіз конкуренції на ринку, SWOT аналіз та обґрунтуванні фактори конкурентоспроможності.

Описаний продукт з використанням представленої технології є доцільним для користувачів, які бажають мати простий інструментарій для роботи із великими даними. Серед них можуть бути як аналітики, що будуть працювати із різними проблемними областями, так і студенти університетів для вивчення алгоритмів машинного навчання у вищих навчальних закладах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kitchin, Rob; McArdle, Gavin "What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets". [Електронний ресурс] — 2016 — Режим доступу до ресурса: https://journals.sagepub.com/doi/10.1177/2053951716631130.
2. Big Data's Fourth V [Електронний ресурс] — 2018 — Режим доступу до ресурса: https://spotlessdata.com/blog/big-datas-fourth-v
3. Виктор Майер-Шенбергер, Кеннет Кукьер. Большие данные. Революция, которая изменит то, как мы живём, работаем и мыслим = Big Data. A Revolution That Will Transform How We Live, Work, and Think / пер. с англ. Инны Гайдюк. — М.: Манн, Иванов, Фербер, 2014. — 240 с. — ISBN 987-5-91657-936-9.
4. First M87 Event Horizon Telescope Results. V. Physical Origin of the Asymmetric Ring [Електронний ресурс] — 2018 — Режим доступу до ресурса: https://iopscience.iop.org/article/10.3847/2041-8213/ab0f43
5. Netflix and Big Data How big data became important to Netflix. [Електронний ресурс] — 2016 — Режим доступу до ресурса: https://www.academia.edu/35644610/Netflix_and_Big_Data
6. Додонов О.Г. Комп'ютерні мережі та аналітичні дослідження / АГ. Додонов, Д.В. Лан- де, В.Г. Путятін. — К.. ИПРИ НАН Украины, 2014 — 486ст.
7. Pyle, D., 1999. Data Preparation for Data Mining. Morgan Kaufmann Publishers, Los Altos, California.
8. Data Mining Curriculum: A Proposal (Version 1.0) [Електронний ресурс] — 2006 — Режим доступу до ресурса: https://www.kdd.org/exploration_files/CURMay06.pdf
9. Методи аналізу даних: навчальний посібник для студентів/ В. Є. Бахрушин. – Запоріжжя: КПУ, 2011. – 268 с. ISBN 978-966-414-103-8

10. Эрик Сигель. Просчитать будущее: Кто кликнет, купит, соврёт или умрёт = Predictive Analytics. — М.: Альпина Паблишер, 2014. — 374 с. — ISBN 978-5-9614-4541-1.
11. The Future of Big Data? Three Use Cases of Prescriptive Analytics [Електронний ресурс] — 2017 — Режим доступу до ресурса: https://vanrijmenam.nl/future-big-data-cases-prescriptive-analytics/
12. Diagnostic Analytics [Електронний ресурс] — 2017 — Режим доступу до ресурса: https://www.gartner.com/en/information-technology/glossary/diagnostic-analytics
13. André B. Bondi, 'Characteristics of scalability and their impact on performance', Proceedings of the 2nd international workshop on Software and performance, Ottawa, Ontario, Canada, 2000, ISBN 1-58113-195-X, pages 195 – 203
14. Bill Venners, Inside the Java Virtual Machine[Електронний ресурс] — 2017 — Режим доступу до ресурса: https://www.artima.com/insidejvm/ed2/jvm.html
15. Learning Malware Analysis. Monnappa K A., — 2018 — 510 pages
16. Tom White Hadoop: The definitive Guide : Storage and Analysis at Internet Scale 4th Edition, USA, 2015, ISBN: 978-1-491-90163-2 pages 754
17. Loverdo, Christos (2010). Steps in Scala: An Introduction to Object-Functional Programming. Cambridge University Press. ISBN 9781139490948.
18. Kafka: The Definitive Guide 2th Edition USA, 2017, ISBN: 978-1-491-99065-0, 332 pages.
19. Michael J. Crawley (2007). The R Book. John Wiley & Sons. ISBN 978-0-470-51024-7.
20. CRAN - Contributed Packages. cran.r-project.org. [Електронний ресурс] Режим доступу до ресурса: https://cran.r-project.org/web/packages/
21. A Survey on Data Science Technologies & Big Data Analytics [Електронний ресурс] — 2016 — Режим доступу до ресурса: http://ijarcsse.com/Before_August_2017/docs/papers/Volume_6/2_February2016/V6I2-0230.pdf

22. Андреас Мюллер, Сара Гвидо. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными = Introduction to Machine Learning with Python: A Guide for Data Scientists. — Вильямс, 2017. — 480 с. — ISBN 978-5-9908910-8-1, 978-1-449-36941-5.
23. Дж. Вандер Плас. Python для сложных задач. Наука о данных и машинное обучение = Python Data Science Handbook: Essential Tools for Working with Data. — Питер, 2017. — 576 с. — ISBN 978-5-496-03068-7.
24. Wes McKinney (2011). "pandas: a Foundational Python Library for Data Analysis and Statistics"
25. Fabian Pedregosa; Gaël Varoquaux; Alexandre Gramfort; Vincent Michel; Bertrand Thirion; Olivier Grisel; Mathieu Blondel; Peter Prettenhofer; Ron Weiss; Vincent Dubourg; Jake Vanderplas; Alexandre Passos; David Cournapeau; Matthieu Perrot; Édouard Duchesnay (2011). "Scikit-learn: Machine Learning in Python". Journal of Machine Learning Research. 12: 2825–2830.
26. Прохоренок Н. А. Python 3 и PyQt. Разработка приложений. — СПб.: БХВ-Петербург, 2012. — С. 704. — ISBN 978-5-9775-0797-4.
27. Джон Уокенбах. Excel 2013: профессиональное программирование на VBA = Excel 2013 Power Programming with VBA. — М.: «Диалектика», 2014. — 960 с. — ISBN 978-5-8459-1877-2.
28. Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON, USA, 2015, ISBN: 1491929480, 126 pages.
29. Дэвид Хантер, Джефф Рафтер, Джо Фаусетт, Эрик ван дер Влист, и др. XML. Работа с XML, 4-е издание = Beginning XML, 4th Edition. — М.: «Диалектика», 2009. — 1344 с. — ISBN 978-5-8459-1533-7.
30. DSV stands for Delimiter Separated Values Raymond, Eric (2004). The Art of Unix Programming. Boston: Addison-Wesley. ISBN 0-13-142901-9.
31. Python Data Science Handbook Essential Tools for Working with Data Jake VanderPlas, USA, 2016, ISBN: 978-1-491-91205-8, 516 pages

32. Optimally splitting cases for training and testing high dimensional classifiers [Електронний ресурс] — 2011 — Режим доступу до ресурса: https://brb.nci.nih.gov/techreport/Dobbin-SampleSplitting.pdf
33. Data Analysis for Omic Sciences: Methods and Applications, Volume 82 1st Edition Joaquim Jaumot Carmen Bedia Roma Tauler, USA, 2018, ISBN: 9780444640451, 740 pages
34. Data preprocessing in detail [Електронний ресурс] — 2019 — Режим доступу до ресурса:https://developer.ibm.com/articles/data-preprocessing-in-detail/
35. Min Max Scaler [Електронний ресурс] — 2018 — Режим доступу до ресурса: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
36. Toby Segaran. Programming Collective Intelligence: Building Smart Web 2.0 Applications. — O'Reilly Media, Inc., 2007. — 308 с. — ISBN 9780596529321. Имеется перевод: Тоби Сегаран. Программируем коллективный разум. — Символ-Плюс, 2009. — 368 с. — ISBN 5-93286-119-3.
37. Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям . — СПб.: Изд. Питер, 2009. — 624 с.
38. Bouckaert, Remco R.; Frank, Eibe; Hall, Mark A.; Holmes, Geoffrey; Pfahringer, Bernhard; Reutemann, Peter; Witten, Ian H. (2010). "WEKA Experiences with a Java open-source project". Journal of Machine Learning Research. 11: 2533–2541.
39. Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic (1996). "From Data Mining to Knowledge Discovery in Databases"
40. Ситник В. Ф., Краснюк М. Т. Інтелектуальний аналіз даних (дейтамайнінг): Навч. посібник. — К.: КНЕУ, 2007. — 376 с
41. Ian H. Witten, Eibe Frank and Mark A. Hall. Data Mining: Practical Machine Learning Tools and Techniques. — 3rd Edition. — Morgan Kaufmann, 2011. — P. 664. — ISBN 9780123748560
42. Hilary L. Seal (1967). "The historical development of the Gauss linear model". Biometrika

43. Tolles, Juliana; Meurer, William J (2016). "Logistic Regression Relating Patient Characteristics to Outcomes". JAMA JAMA. 316 (5): 533. ISSN 0098-7484. OCLC 6823603312.
44. What is Naive Bayes in Machine Learning [Електронний ресурс] — 2019 — Режим доступу до ресурса: https://www.knowledgehut.com/blog/data-science/naive-bayes-in-machine-learning
45. Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks"
46. Leo Breiman Machine Learning journal. — 2001. — Vol. 45, no. 1. — P. 5—32.
47. Случайный лес (Random Forest) [Електронний ресурс] — 2016 — Режим доступу до ресурса: https://dyakonov.org/2016/11/14/случайный-лес-random-forest/
48. Tolosi L., Lengauer T. Classification with correlated features: unreliability of feature ranking and solutions.
49. Разработка продуктовой стратегии компании [Електронний ресурс] — 2019 — Режим доступу до ресурса: https://pidruchniki.com/19670511/marketing/razrabotka_produktovoy_strategii_kompanii
50. Маркетингове поняття та класифікація видів товару [Електронний ресурс] — 2019 — Режим доступу до ресурса: http://marketing-helping.com/konspekti-lekcz/21-konspekt-lekczj- qosnovi-marketinguq/412-marketingove-ponyattya-ta-klasifkaczya-vidv-tovaru.html

ДОДАТОК 1

Інтсрументальні засоби моделювання сценаріїв аналітики великих даних

Апробація

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ ТР4153_18Б 12-1

Аркушів 5

2019

ІНТСРУМЕНТАЛЬНІ ЗАСОБИ ПОБУДОВИ СЦЕНАРІЇВ АНАЛІТИКИ ВЕЛИКИХ ДАНИХ В ІНФОРМАЦІЙНО-АНАОІТИЧНИХ СИСТЕМАХ

Інформаційно-аналітична система (ІАС) – це окремий клас інформаційних систем, що призначені для аналітичної обробки даних, яка об’єднує, аналізує і зберігає інформацію, видобуту як з баз даних організації, так і із зовнішніх джерел[1, с. 124].

Подібні системи створюються, як надбудова над уже імплементованим на підприємстві програмами, без необхідності внесення будь яких змін до їх функціоналу. Основним призначенням систем цього класу є забезпечення адміністрації, відділів аналітики та менеджерів компанії інформацією про всі ключові аспекти діяльності підприємства для її подальшої оцінки та аналізу. До складу інформаційно-аналітичної системи зазвичай входять наступні модулі:

- модуль збору і зберігання корпоративних даних, завдання якого збір та фільтрація даних, накопичення і індексування інформації, що забезпечує можливість використання отриманих знань в аналітичних цілях, а також за підтримки прийняття рішень на усіх рівнях управління;
- модуль доступу до даних, аналізу та корпоративної звітності, що забезпечує доступ до даних і захист конфіденційної інформації, інструментарій з підтримки прийняття рішень, і своєчасного аналізу інформації, засоби детальної звітності, візуалізації даних та побудови можливих сценаріїв на основі збережених даних.

Модуль побудови сценарію — важлива складова, яка повинна виконувати дві основні функції. По-перше, надавати весь необхідний функціонал для можливості створювати сценарії будь-якої складності для прогнозів діяльності підприємства. По-друге, графічний інтерфейс із функціоналом drag-and-drop (перетягування) повинен бути простим та зрозумілим для кінцевого користувача.

В даній роботі розглядаються наступне програмне забезпечення із відкритим кодом, що можна використовувати при побудові інформаційно-аналітичної системи як модуль побудови сценарію:

- **Rapid Miner** — це кросплатформене програмне забезпечення, розроблене компанією з однойменною назвою. Rapid Miner забезпечує середовище для машинного навчання, інтелектуального аналізу даних та інтелектуального аналізу тексту. Програмне забезпечення використовується, як у компаніях для аналітики великих бізнес процесів, так і для наукових дослідженнях, навчання студентів, швидкого створення прототипів і розробки додатків. Rapid Miner розроблений на основі клієнт-серверної моделі.
- **Orange**— це компонентний набір кросплатформеного програмного забезпечення для інтелектуального аналізу даних і машинного навчання, що містить інтерфейс візуального програмування для дослідження та візуалізації даних. Розвиток програмного продукту розпочався в 1997 році в лабораторії біоінформатики в Люблінському університеті.
- **KNIME** — це кросплатформене програмне забезпечення для аналізу даних, звітності та інтеграції з відкритим вихідним кодом, розроблена та підтримувана однойменною компанією. Розвиток KHIME розпочався у січні 2004 року командою програмних інженерів Університету м. Констанц. Початковою метою було створення модульної, високо масштабованої та відкритої платформи інтелектуальної обробки даних не орієнтуючись на якусь конкретну область застосування.

Кожен аналітичний сценарій складається із декількох операторів. Оператор виконує одне спеціальне завдання в межах сценарію, а вихід кожного оператора формує данні, що будуть надані на вхід іншого. Процес побудови сценарію можна поділити на три основні етапи:

- Зчитування даних із різних джерел;
- Обробка та робота із даними;
- Візуалізація та експорт отриманого результату.

Саме за цією структурою буде розглянутий фунціонал вище наведеного програмного забезпечення.

На великих підприємствах із десятками, а то й сотнями філій встановлюють різне програмне забезпечення, яке зберігає інформацію у різні формати даних.

Основна вимога для модуля побудови сценарію на даному етапі — забезпечити користувача можливістю завантажити усю необхідну інформацію для побудови конкретного сценарію.

- **Rapid Miner.** Для роботами з різними джерелами даних у Rapid Miner існує близько 53 операторів. Серед яких є робота із файлами різного формату (csv, excel, ARFF, XRFF, SPSS), базами даних, програмним забезпеченням (Salesforce, Twitter), хмарними сховищами (Amazon, Google, Azure) та локальними репозиторіями. Проте в безкоштовній версії Rapid Miner можливо працювати лише із набором даних, кількість рядків у яких не перевищує 10 000. Більш того не можна використовувати технологію Radoop (інтеграції файлової системи Hadoop у Rapid Miner);
- **Orange.** Для роботи із даними в Orange за замовчуванням є три оператори: оператор “File” підтримує файли Excel та csv формату, оператор “Datasets” підтримує дані, що були завантажені з онлайн репозиторіїв. Оператор “SQL Table” дозволяє завантажити дані із серверу. Для роботи із зображення, текстовими файли тощо необхідно встановлювати додаткові оператори через вкладку “Addons”.
- **KNIME** Для роботи із даними в KNIME може працювати із xlsx, csv, ARFF, XML, JSON, SQL Database, а також наявна можливо отримувати дані із REST сервісів.

Важливою складовою модуля побудови сценарію це обробка даних, які були зібрані із різних джерел інформації. Даний етап можна поділити на дві під задачі: попередня обробка даних та побудова прогнозу. Основна вимога до фунціоналу модуля побудови сценарію — надати необхідну кількість інструментів для обох задач.

- **Rapid Miner.** Серед представленого в даній роботі програмного забезпечення функціонал Rapid Miner є одним із найкращих. Для процесу агрегації та об’єднання наборів данних представлено більше ніж 76 операторів серед яких є фільтрування, оператори join, merge тощо. Для процесу Data cleaning (очистки даних) надається 25 операторів. Цей фунціонал повністю закриває першу під задачу. Для побудови

прогнозу, тобто моделювання надається більше 153 оператори, серед яких різні типи регресій (логістичні, лінійні) та класифікацій;

- **Orange.** Функціонал Orange за замовчуванням помітно менший ніж у Rapid Miner. Оператор Preprocess нормалізую данні, опрацьовує рядки із відсутніми значеннями, тощо . Для побудови прогнозу є всього 16 операторів, які тим не менш покривають основний мінімум — є лінійна та логістична регресії, оператор нейронної мережі, що використовує мультишаровий персептрон, тощо;
- **KNIME.** Надаваємий KNIME функціонал покриває задачу попередньої обробки даних, хоча кількість операторів менша ніж у Rapid Miner. Проте операторів для прогнозу достатньо, щоб побудувати сценарій високої складності.

Останнім етапом у процесі побудові сценарію — візуалізація та експорт отриманого результату. Інформаційно-аналітична система повинна давати змогу зберігати прогнози у декількох форматах та будувати різні графічні представлення.

- **Rapid Miner.** Дане програмне забезпечення надає можливість експортувати в файли різного формату, бази даних та хмарні сховища. Представити результат можна у 28 графічних форматів із різними налаштуванням, які можна зберігати у форматі jpg, png та pdf.
- **Orange.** Функціонал Orange надає можливість зберігати результати у файлах 3-ьох типів: csv, excel та текстовий файл. Orange надає за замовчуванням 21 графік, які можна зберігати в jpg форматі.
- **KNIME.** Результати KNIME може записати в ті самі формати, що й зчитувати, а також у бази даних. 16 графіків можна зберігати у jpg форматі.

Rapid Miner серед усіх конкурентів має найбільшу кількість операторів. Проте через ліміти безкоштовної версії розглядати дане ПО для використання на середніх та великих підприємствах без додаткових грошових вливань не можна. Те саме можна сказати і про KNIME, оскільки доступ до REST API можливий лише через платну версію. Orange, хоча й має скромніший функціонал, повністю підходить умовам для модулю побудови сценарію і є повністю безкоштовним програмним забезпеченням.